

VIDYAPITH ACADEMY

A unit of **AITDC (OPC) PVT. LTD.**

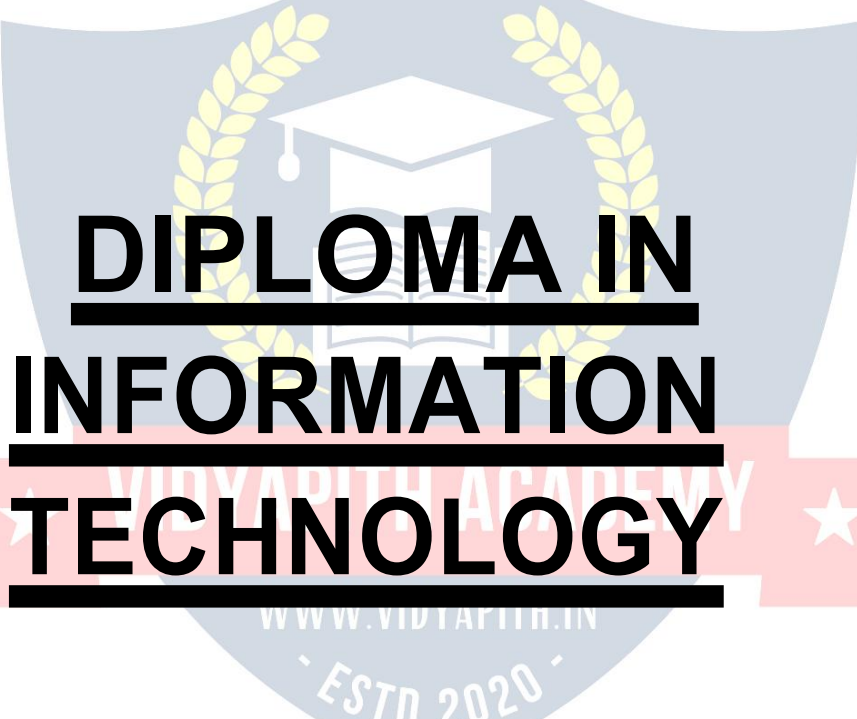
IAF Accredited An ISO 9001:2015 Certified Institute.

Registered Under Ministry of Corporate Affairs

(CIN U80904AS2020OPC020468)

Registered Under MSME, Govt. of India. (UAN- AS04D0000207).

Registered Under MHRD (CR act) Govt. of India.



TOPIC 1: COMPUTER FUNDAMENTALS

TOPIC 2: HTML/DHTML, DREAMWEAVER

TOPIC 3: PROGRAMMING LANGUAGE C

TOPIC 4: PROGRAMMING LANGUAGE C++

**TOPIC 5: PROGRAMMING IN JAVA PRACTICAL
LAB ASSIGNMENT & VIVA VOICE**

COMPUTER FUNDAMENTALS

Computer is an electronic device that is designed to work with Information. The term computer is derived from the Latin term 'compute', this means to calculate. Computer cannot do anything without a Program. It represents the decimal numbers through a string of binary digits. The Word 'Computer' usually refers to the Center Processor Unit plus Internal memory.

CHARLES BABBAGE

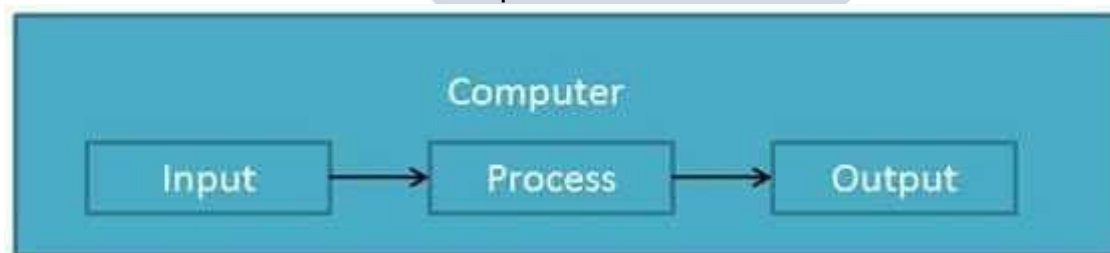
Charles Babbage, was an English polymath. He was a mathematician, philosopher, inventor and mechanical engineer, who is best remembered now for originating the concept of a programmable computer.

Considered a "father of the computer", Babbage is credited with inventing the first mechanical computer that eventually led to more complex designs. His varied work in other fields has led him to be described as "pre-eminent" among the many polymaths of his century.

FUNCTIONALITES OF COMPUTER:

Any digital computer carries out five functions in gross terms:

- Takes data as input.
- Stores the data/instructions in its memory and can use them when required.
- Process the data and convert it into useful information.
- Output the information.
- Control all the above four steps.



Definition

Computer System is an electronic data processing device which does the following:

- Accept and store an input data.
- Process the data input.

- And output the processed data in required format.

ADVANTAGES OF COMPUTER:

Following list demonstrates the advantages of Computers : **High Speed**

- Computer is a very fast device.
- It is capable of performing addition of very big data.
- The computer has units of speed in microsecond, nanosecond and even the picoseconds.
- It can perform millions of calculations in a few seconds as compared to man who can spend many months for doing the same task.

Accuracy

- In addition to being very fast, computer is very accurate. □ The computer can perform calculations 100% error free.
- Computers perform all jobs with 100% accuracy.

Storage Capability

- Memory is a very important characteristic of a computer.
- The computer has much more storage capacity than human beings.
- It can store large amount of data.
- It can store any type of data such as images, videos, text, audio and any other type.

Diligence

- Unlike human beings, a computer is free from monotony, tiredness and lack of concentration.
- It can work continuously without creating any error and boredom.
- It can do repeated work with same speed and accuracy.

Versatility

- A computer is a very versatile machine.
- A computer is very flexible in performing the jobs to be done.
- This machine can be used to solve the problems relating to various different fields.
- At one instant, it may be solving a complex scientific problem and the very next moment it may be playing a card game. **Reliability**
- A computer is a reliable machine.
- Modern electronic components have failure free long lives.
- Computers are designed to make maintenance easy.

Automation

- Computer is an automatic machine.
- Automation means ability to perform the task automatically.
- Once a program is given to computer i.e stored in computer memory, the program and instruction can control the program execution without human interaction.

Reduction In Paper Work

- The use of computers for data processing in an organization leads to reduction in paper work and speeds up the process.
- As data in electronic files can be retrieved as and when required, the problem of maintenance of large number of files gets reduced.

Reduction In Cost

- Though the initial investment for installing a computer is high but it substantially reduces the cost of each of its transaction.

DISADVANTAGES OF COMPUTER:

Following list demonstrates the disadvantages of Computers in today's arena.

- A computer is a machine and has no intelligence of its own to perform any task.
- Each and every instruction has to be given to the computer.

- A computer cannot take any decision on its own.

Dependency

- It can perform function as instructed by the user, so it is fully dependent on human being.

Environment

- The operating environment of computer should be dust free and suitable to it. **No Feeling**
- Computer has no feeling or emotions.
- It cannot make Judgment based on feelings, tastes, experiences and knowledge unlike a human being.

APPLICATIONS OF COMPUTER

Banking

Today Banking is almost totally dependent on computer.

Banks provide following facilities:

- Banks on-line accounting facility, which include current balances, deposits, overdrafts, interest charges, shares and trustee records.
- ATM machines are making it even easier for customers to deal with banks.

Insurance

Insurance companies are keeping all records up to date with the help of computer. The Insurance Companies, Finance houses and Stock broking firms are widely using computers for their concerns.

Insurance Companies are maintaining a database of all clients with information showing

- how to continue with policies
- starting date of the policies
- next due installment of a policy
- maturity date
- interests due
- survival benefits

- bonus

Education

The computer has provided a lot of facilities in the Education System.

- The uses of computer provide a tool in the Education system is known as CBE (Computer Based Education).
- CBE involves Control, Delivery and Evaluation of learning.
- The computer education is very familiar and rapidly increasing the graph of computer students.
- There are number of methods in which educational institutions can use computer to educate the students.
- It is used for prepare a database about student performance and analysis are carried out.

Marketing

In Marketing uses of computer are following:

- **Advertising:** With computers, advertising professionals create art and graphics, write and revise copy and print and disseminate ads with the goal of selling more products.
- **At Home Shopping:** At home shopping has been made possible through use of computerized catalogues that provide access to product information and permit direct entry of orders to be filled by the customers.

Health Care

Computers have become important part in all Medical Systems.

The computers are being used in hospitals to keep the record of patients and medicines. It is also used in scanning and diagnosing different diseases. ECG, EEG, Ultrasounds and CT Scans etc. are also done by computerized machines.

Some of major fields of health care in which computer are used:

- **Diagnostic System:** Computers are used to collect data and identify cause of illness.
- **Lab-diagnostic System:** All tests can be done and reports are prepared by computer.
- **Patient Monitoring System:** These are used to check patient's signs for abnormality such as in Cardiac Arrest, ECG etc.
- **Pharma Information System:** Computer checks Drug-Labels, Expiry dates, harmful drug side effects etc.
- Nowadays, computers are also used in performing surgery.

Engineering Design

Computers are widely used in Engineering purposes.

One of major areas is CAD (Computer Aided Design). CAD provides creation, edition and modification of image. Some fields are:

- **Structural Engineering:** Requires stress and strain analysis required for design of Ships, Buildings, Budgets, and Airplanes etc.
- **Industrial Engineering:** Computers deals with design, implementation and improvement of integrated systems of people, materials and equipments.
- **Architectural Engineering:** Computers help in planning towns, designing buildings, determining a range of buildings on a site using both 2D and 3D drawings.

Military

Computers are largely used in defense. Modern tanks, missiles, weapons etc. employ computerized control systems. Some military areas where a computer has been used are:

- Missile Control
- Military Communication
- Military operation and planning
- Smart Weapons

Communication

Communication means to convey a message, an idea, a picture or speech that is received and understood clearly and correctly by the person for whom it is meant. Some main areas in this category are:

- E-mail
- Chatting
- Usenet
- FTP
- Telnet
- Video-conferencing

Government applications

Computers play an important role in government applications. Some major fields in this category are:

- Budgets
- Sales tax department
- Income tax department
- Male/Female ratio
- Computerization of voters lists
- Computerization of Driving Licensing system
- Computerization of PAN card
- Weather Forecasting.

TYPES OF COMPUTER:

Computer can be broadly classified by their speed and computing power.

Sr. No.	Type	Specifications
1	PC (Personal Computer)	Single user computer system. Moderately powerful microprocessor.
2	WorkStation	Single user computer system. Similar to Personal Computer but have more powerful microprocessor.
3	Mini Computer	Multi-user computer system. Capable of supporting hundreds of users simultaneously.
4	Main Frame	Multi-user computer system. Capable of supporting hundreds of users simultaneously. Software technology is different from minicomputer.
5	Supercomputer	An extremely fast computer which can perform hundreds of millions of instructions per second.

INPUT DEVICES

Following are few of the important input devices which are used in Computer Systems

- Keyboard
- Mouse
- Joy Stick
- Light pen
- Track Ball
- Scanner
- Graphic Tablet
- Microphone
- Magnetic Ink Card Reader(MICR)
- Optical Character Reader(OCR)
- Bar Code Reader
- Optical Mark Reader

OUTPUT DEVICES

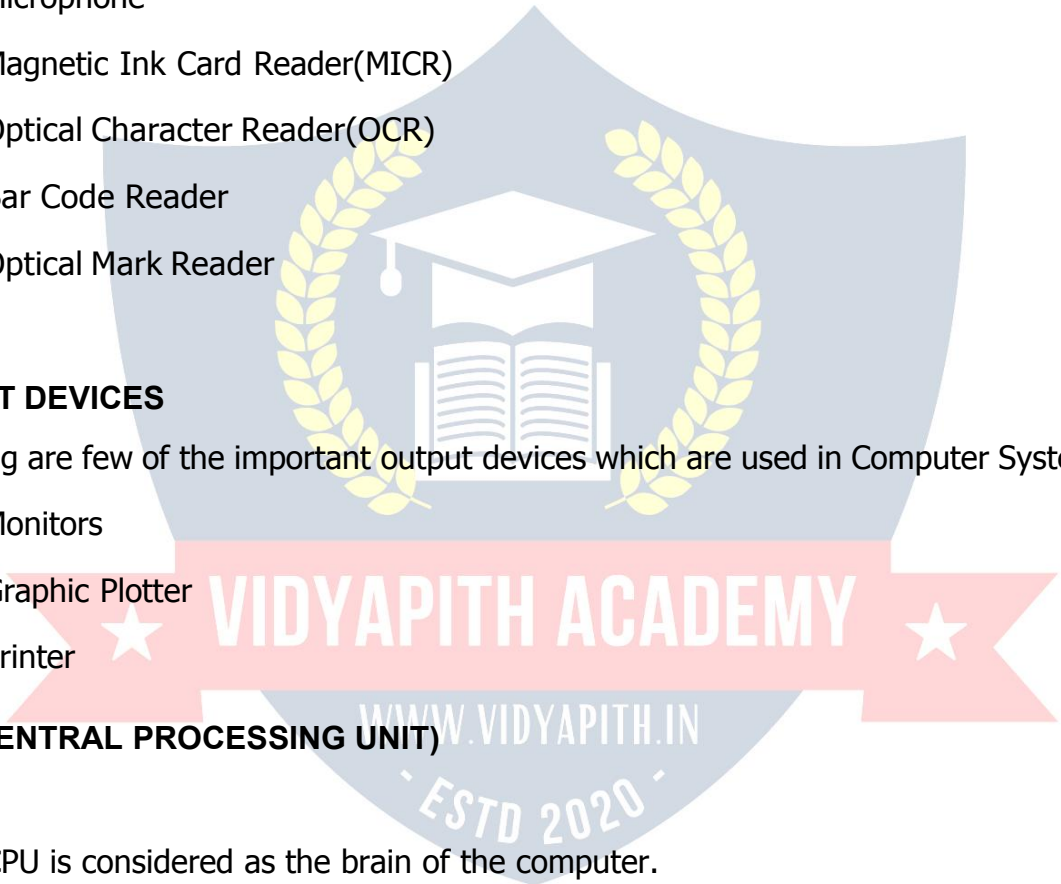
Following are few of the important output devices which are used in Computer Systems

- Monitors
- Graphic Plotter
- Printer

CPU (CENTRAL PROCESSING UNIT)

- CPU is considered as the brain of the computer.
- CPU performs all types of data processing operations.
- It stores data, intermediate results and instructions (program).
- It controls the operation of all parts of computer.

CPU itself has following three components.



- Memory Or Storage Unit:
- Control Unit
- ALU (Arithmetic Logic Unit)

Memory or Storage Unit:

This unit can store instruction, data and intermediate results. This unit supplies information to the other units of the computer when needed. It is also known as internal storage unit or main memory or primary storage or Random access memory (RAM).



Its size affects speed, power and capability. There are primary memory and secondary memory two types of memories in the computer.

Function of Memory Unit is:

It stores all the data to be processed and the instructions required for processing.

It stores intermediate results of processing.

It stores final results of processing before these results are released to an output device.

All inputs and outputs are transmitted through main memory.

Control Unit:

This unit controls the operations of all parts of computer. It does not carry out any actual data processing operations.

Functions of this unit are

It is responsible for controlling the transfer of data and instructions among other units of a computer.

It manages and coordinates all the units of the computer.

It obtains the instructions from the memory, interprets them and directs the operation of the computer.

It communicates with Input/output devices for transfer of data or results from storage.

It does not process or store data.

ALU (Arithmetic Logic Unit):

This unit consists of two sub-sections namely

- Arithmetic section
- Logic Section

ARITHMETIC SECTION

Function of Arithmetic section is to perform arithmetic operations like addition, subtraction, multiplication and division. All complex operations are done by making repetitive use of above operations.

LOGIC SECTION

Function of logic section is to perform logic operations such as comparing, selecting, matching and merging of data.

Units of Computer Memory Measurements

1 Bit = Binary Digit (Bits of Computer are 0 and 1)

8 Bits = 1 Byte

1024 Bytes = 1 KB (Kilo Byte)

1024 KB = 1 MB (Mega Byte)

1024 MB = 1 GB(Giga Byte)

Computer Memory

In computing, **memory** refers to the physical devices used to store programs (sequences of instructions) or data (e.g. program state information) on a temporary or permanent basis for use in a computer or other digital electronic device. The term primary memory is used for the information in physical systems which function at highspeed (i.e. RAM), as a distinction from secondary memory, which are physical devices for program and data storage which are slow to access but offer higher memory capacity. Primary memory stored on secondary memory is called "virtual memory". An archaic synonym for memory is **store**.

The term "memory", meaning primary memory is often (but not always) associated with addressable semiconductor memory, i.e. integrated circuits consisting of silicon-based transistors, used for example as primary memory but also other purposes in computers and other digital electronic devices. There are two main types of semiconductor memory: volatile and non-volatile. Examples of non-volatile memory are flash memory (sometimes used as secondary, sometimes primary computer memory) and ROM/PROM/EPROM/EEPROM memory (used for firmware such as boot programs). Examples of volatile memory are primary memory (typically dynamic RAM, DRAM), and fast CPU cache memory (typically static RAM, SRAM, which is fast but energyconsuming and offer lower memory capacity per area unit than DRAM) .

Volatile memory

Volatile memory is computer memory that requires power to maintain the stored information. Most modern semiconductor volatile memory is either Static RAM or dynamic RAM. SRAM retains its contents as long as the power is connected and is easy to interface to but uses six transistors per bit. Dynamic RAM is more complicated to interface to and control and needs regular refresh cycles to prevent its contents being lost. However, DRAM uses only one transistor and a capacitor per bit, allowing it to reach much higher densities and, with more bits on a memory chip, be much cheaper per bit. SRAM is not worthwhile for desktop system memory, where DRAM dominates, but is used for their cache memories. SRAM is commonplace in small embedded systems, which might only need tens of kilobytes or less. Forthcoming volatile memory technologies that hope to

replace or compete with SRAM and DRAM include Z-RAM, TTRAM, A-RAM and ETA RAM.

Non-volatile memory

Non-volatile memory is computer memory that can retain the stored information even when not powered. Examples of non-volatile memory include read-only memory(ROM), flash memory, most types of magnetic computer storage devices (e.g. hard disks, floppy discs and magnetic tape), optical discs, and early computer storage methods such as paper tape and punched cards.

OPERATING SYSTEM

An operating system is a program that acts as an interface between the software and the computer hardware.

- It is an integration set of specialized programs that are used to manage overall resources and operations of the computer.
- It is specialized software that controls and monitors the execution of all other programs that reside in the computer, including application programs and other system software.

Objectives of Operating System

- Making a computer system convenient to use in an efficient manner
- To hide the details of the hardware resources from the users
- To provide users a convenient interface to use the computer system.
- To act as an intermediary between the hardware and its users and making it easier for the users to access and use other resources.
- Manage the resources of a computer system.
- Keep track of who is using which resource, granting resource requests, according for resource using and mediating conflicting requests from different programs and users.
- The efficient and fair sharing of resources among users and programs

Characteristics of Operating System

- **Memory Management** -- It keeps tracks of primary memory i.e what part of it are in use by whom, what part are not in use etc. Allocates the memory when the process or program request it.

- **Processor Management** -- Allocate the processor(CPU) to a process. De-allocate processor when processor is no longer required.
- **Device Management** -- Keep tracks of all devices. This is also called I/O controller. Decides which process gets the device when and for how much time.
- **File Management** -- Allocates the resources. De-allocates the resources. Decides who gets the resources.
- **Security** -- By means of passwords & similar other techniques, preventing unauthorized access to programs & data.
- **Job accounting** -- Keeping track of time & resources used by various jobs and/or users.
- **Control over system performance** -- Recording delays between request for a service & from the system.
- **Interaction with the operators** -- The interaction may take place via the console of the computer in the form of instructions. Operating System acknowledges the same, do the corresponding action and inform the operation by a display screen.
- **Error-detecting aids** -- Production of dumps, traces, error messages and other debugging and error-detecting methods.
- **Coordination between other software and users** -- Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

WINDOWS 7

Windows 7 is an operating system produced by Microsoft for use on personal computers, including home and business desktops, laptops, net books, tablet PCs, and media center PCs. It was released to manufacturing on July 22, 2009, and became generally available retail worldwide on October 22, 2009, less than three years after the release of its predecessor, Windows Vista. Windows 7's server counterpart, Windows Server 2008 R2, was released at the same time. Windows 7 is succeeded by Windows 8.

Unlike Windows Vista's many new features, Windows 7 was an incremental upgrade designed to work with Vista-compatible applications and hardware. Presentations given by Microsoft in 2008 focused on multi-touch support, an updated Windows shell with a new taskbar, referred to internally as the *Superbar*, a home networking system called Home Group, and performance improvements. Some standard applications that have been included with prior releases of Microsoft Windows, including Windows Calendar, Windows Mail, Windows Movie Maker, and Windows Photo Gallery, are not included in Windows 7; most are instead offered separately at no charge as part of the Windows Essentials suite.

Install Windows 7

Many people have computers that come with Windows 7 so they may never have to install it. However, you may need to install Windows 7 if:

- You replaced your hard disk drive with a new hard disk drive that does not have Windows 7 installed.
- You are reinstalling Windows 7 on a computer because you want to clean off your hard drive and remove any unwanted programs, such as spyware.
- You purchased a computer without an operating system.

Pre-installation checklist

Before you begin the installation process, use this checklist to make sure that you are prepared:

- You have the Windows 7 CD.
- You have the product key available. The product key is located on your Windows 7 CD case and is required to install and activate Windows 7.
- Your computer hardware is set up. At a minimum, you should connect your keyboard, mouse, monitor, and CD drive. If available, you should connect your computer to a wired network.
- You have Windows 7 drivers available. Drivers are software that Windows 7 uses to communicate with your computer's hardware. If you do not have drivers available, Windows 7 may already include drivers for your hardware. If not, you should be able to download them from your hardware manufacturer's website after you set up Windows 7.
- If you are reinstalling Windows 7 on an existing computer, you need a backup copy of your files and settings. The installation process will delete all of your files. You can use the File and Settings Transfer Wizard to store your files and settings on removable media and then restore them after installation is complete.

Installation process

Installing Windows 7 can take up to two hours. To make the process more manageable, it has been broken up into several sections.

To Begin Installation:

1. Insert the Windows 7 CD into your computer and restart your computer.
2. If prompted to start from the CD, press Spacebar. If you miss the prompt (it only appears for a few seconds), restart your computer to try again.
3. Windows 7 Setup begins. During this portion of setup, your mouse will not work, so you must use the keyboard. On the Welcome to Setup page, press Enter.
4. On the Windows 7 Licensing Agreement page, read the licensing agreement. Press the Page Down key to scroll to the bottom of the agreement. Then press F8.
5. This page enables you to select the hard disk drive on which Windows 7 will be installed. Once you complete this step, all data on your hard disk drive will be removed and cannot be recovered. It is extremely important that you have a recent backup copy of your files before continuing. When you have a backup copy, press D, and then press L when prompted. This deletes your existing data.

6. Press Enter to select unpartitioned space, which appears by default.
7. Press Enter again to select Format the partition using the NTFS file system, which appears by default.
8. Windows 7 erases your hard disk drive using a process called formatting and then copies the setup files. You can leave your computer and return in 20 to 30 minutes.

To Continue the Installation

1. Windows 7 restarts and then continues with the installation process. From this point forward, you can use your mouse. Eventually, the Regional and Language Options page appears. Click Next to accept the default settings. If you are multilingual or prefer a language other than English, you can change language settings after setup is complete.
2. On the Personalize Your Software page, type your name and your organization name. Some programs use this information to automatically fill in your name when required. Then, click next.
3. On the Your Product Key page, type your product key as it appears on your Windows 7 CD case. The product key is unique for every Windows 7 installation. Then, click Next.
4. On the Computer Name and Administrator Password page, in the Computer name box, type a name that uniquely identifies your computer in your house, such as FAMILYROOM or TOMS. You cannot use spaces or punctuation. If you connect your computer to a network, you will use this computer name to find shared files and printers. Type a strong password that you can remember in the Administrator password box, and then retype it in the Confirm password box. Write the password down and store it in a secure place. Click Next.
5. On the Date and Time Settings page, set your computer's clock. Then, click the Time Zone down arrow, and select your time zone. Click Next.
6. Windows 7 will spend about a minute configuring your computer. On the Networking Settings page, click next.
7. On the Workgroup or Computer Domain page, click Next.

To complete the installation

1. Windows 7 will spend 20 or 30 minutes configuring your computer and will automatically restart when finished. When the Display Settings dialog appears, click OK.
2. When the Monitor Settings dialog box appears, click OK.
3. The final stage of setup begins. On the Welcome to Microsoft Windows page, click Next.
4. On the Help protect your PC page, click Help protect my PC by turning on Automatic Updates now. Then, click Next.
5. Windows 7 will then check if you are connected to the Internet:
 - o If you are connected to the Internet, select the choice that describes your network connection on the Will this computer connect to the Internet

- directly, or through a network page. If you're not sure, accept the default selection, and click Next.
- If you use dial-up Internet access, or if Windows 7 cannot connect to the Internet, you can connect to the Internet after setup is complete. On the How will this computer connect to the Internet? page, click Skip.
6. Windows 7 Setup displays the Ready to activate Windows page. If you are connected to the Internet, click Yes, and then click Next. If you are not yet connected to the Internet, click No, click Next, and then skip to step 24. After setup is complete, Windows 7 will automatically remind you to activate and register your copy of Windows 7.
 7. On the Ready to register with Microsoft page, click Yes, and then click Next.
 8. On the Collecting Registration Information page, complete the form. Then, click Next.
 9. On the Who will use this computer page, type the name of each person who will use the computer. You can use first names only, nicknames, or full names. Then click Next.
 10. On the Thank you! Page, click Finish.

Windows 7 setup is complete. You can log on by clicking your name on the logon screen. If you've installed Windows 7 on a new computer or new hard disk drive, you can now use the File and Settings Transfer Wizard to copy your important data to your computer or hard disk drive.

MS-DOS

Short for **Microsoft Disk operating system**, **MS-DOS** is a non-graphical command line operating system derived from 86-DOS that was created for IBM compatible computers. MS-DOS originally written by Tim Peterson and introduced by Microsoft in August 1981 and was last updated in 1994 when MS-DOS 6.22 was released. Today, MS-DOS is no longer used; however, the command shell, more commonly known as the **Windows command line** is still used by many users.

Today, most computer users are only familiar with how to navigate Microsoft Windows using the mouse Unlike Windows, MS-DOS is a command-line and is navigated by using MS-DOS commands. For example, if you wanted to see all the files in a folder in Windows you would double-click the folder to open the folder in Windows Explorer. In MS-DOS, to view that same folder you would navigate to the folder using the CD command and then list the files in that folder using the dir command.

DOS Commands

MS-DOS has a relatively small number of commands, and an even smaller number of commonly used ones. Moreover, these commands are generally inflexible because, in contrast to Linux and other Unix-like operating systems, they are designed to accommodate few options or *arguments* (i.e., values that can be passed to the commands).

Some of the most common commands are as follows (corresponding commands on Unix-like operating systems are shown in parenthesis):

CD - changes the current directory (cd)

COPY - copies a file (cp)

DEL - deletes a file (rm)

DIR - lists directory contents (ls)

EDIT - starts an editor to create or edit plain text files (vi, vim, ed, Joe)

FORMAT - formats a disk to accept DOS files (mformat)

HELP - displays information about a command (man, info)

MKDIR - creates a new directory (mkdir)

RD - removes a directory (rmdir)

REN - renames a file (mv)

TYPE - displays contents of a file on the screen (more, cat)

Other DOS Command commonly used are:

Append

The append command can be used by programs to open files in another directory as if they were located in the current directory.

Attrib

The attrib command is used to change the attributes of a single file or a directory.

Break

The break command sets or clears extended CTRL+C checking.

Call

The call command is used to run a script or batch program from within another script or batch program.

The call command has no effect outside of a script or batch file. In other words, running the call command at the DOS prompt will do nothing.

Chcp

The chcp command displays or configures the active code page number.

Chdir

The chdir command is used to display the drive letter and folder that you are currently in.

Chdir can also be used to change the drive and/or directory that you want to work in.

Chkdsk

The chkdsk command, often referred to as *check disk*, is used to identify and correct certain hard drive errors.

Choice

The choice command is used within a script or batch program to provide a list of choices and return the value of that choice to the program.

Cls

The cls command clears the screen of all previously entered commands and other text.

Dir

The dir command is used to display a list of files and folders contained inside the folder that you are currently working in.

The dir command also displays other important information like the hard drive's serial number, the total number of files listed, their combined size, the total amount of free space left on the drive, and more.

DIR [drive:][path][filename] [/P] [/W] [/A[:]attributes] [/O[:]sort order] [/S] [/B] [/L] [/V]

[drive:][path][filename]	Specifies drive, directory, or files to list. (Could be enhanced file specification or multiple file specs)
/P	Pauses after each screenful of information.
/W	Uses wide list format.
/A	attributes: D Directories R Read-only files H Hidden files A Files ready for archiving S System files - Prefix meaning not
/O	List by files in sorted order, sort order: N By name (alphabetic) S By size (smallest first) E By extension (alphabetic) D By date and time (earliest first) G Group directories first - Prefix to reverse order A By Last Access Date (earliest first)
/S	Displays files in specified directory and all subdirectories.
/B	Uses bare format (no heading information or summary).
/L	Uses lowercase.
/V	Verbose mode.

WINDOWS OPERATIONS:

MS- PAINT

Paint is a drawing tool you can use to create simple or elaborate drawings. These drawings can be either black-and-white or color, and can be saved as bitmap files. You can print your drawing, use it for your desktop background, or paste it into another document. You can even use Paint to view and edit scanned photos.


You can also use Paint to work with pictures, such as .jpg, .gif, or .bmp files. You can paste a Paint picture into another document you've created, or use it as your desktop background.

To draw a straight line


1. In the toolbox, click **Line** .

2. Below the toolbox, click a line width.
3. Drag the pointer to draw the line.

To draw a freeform line

1. In the toolbox, click **Line** .
2. Drag the pointer to draw the line.

To draw a curved line



1. In the toolbox, click **Curve** .
2. Below the toolbox, click a line width.
3. Draw a straight line by dragging the pointer.
4. Click where you want one arc of the curve to be, and then drag the pointer to adjust the curve. Repeat this step for a second arc. You can only create two curves for each line.

To send an image using e-mail

After you are done working on an image, you can send it to anyone who has an e-mail account (as long as you also have access to an e-mail account).

1. On the **File** menu, click **Send**.
2. Your e-mail program will display, with your image included either as an attachment or as part of the message section of your e-mail.
3. Specify your recipient's e-mail address, a subject, and a message; then send the email as you normally would.

To draw a rectangle or square

1. In the toolbox, click **Rectangle**  to create a square-cornered shape, or click **Rounded Rectangle**  to create a round-cornered shape.
2. Below the toolbox, click a fill style.
3. To draw a rectangle, drag the pointer diagonally in the direction you want.
4. To draw a square, hold down SHIFT while dragging the pointer.

To acquire an image from a scanner or a digital camera 1.

On the **File** menu, click **From Scanner or Camera**.

2. In the **Get Pictures** dialog box, click the image you want, and then click **Get Picture**.

NOTEPAD

Notepad overview

Notepad is a basic text editor you can use for simple documents or for creating Web pages. The most common use for Notepad is to view or edit text (.txt) files, but many users find Notepad a simple tool for creating Web pages.

Because Notepad supports only very basic formatting, you cannot accidentally save special formatting in documents that need to remain pure text. This is especially useful when creating HTML documents for a Web page because special characters or other formatting may not appear in your published Web page or may even cause errors. You can save your Notepad files as Unicode, ANSI, UTF-8, or big-endian Unicode. These formats provide you greater flexibility when working with documents that use different character sets.

WORDPAD

WordPad is a basic word processor that is included with almost all versions of Microsoft Windows from Windows 95 onwards. It is more advanced than Notepad but simpler than Microsoft Works Word Processor and Microsoft Word. It replaced Microsoft Write.

Using WordPad

WordPad is a basic word processor that is included in Windows. A word processor is a computer program that you can use to create, edit, view, and print text documents. With WordPad, you can type letters, book reports, and other simple documents. You can also change how the text looks, quickly move sentences and paragraphs around, and copy and paste text within and between documents.

The WordPad window

To open WordPad, click the **Start** button , click **All Programs**, click **Accessories**, and then click **WordPad**.

There are four main parts of the WordPad window:

The toolbar: The toolbar contains buttons for basic commands, including saving and printing. To find out what a button does, rest your mouse pointer on it. A box pops up showing the function of the button.

The format bar: The format bar contains buttons that you can use to format the text in your document. For example, you can choose the font, color, and alignment of your text.

The ruler: Use the ruler to check the layout and placement of text in your document.

The document area: This is where you type your document, make changes to the text, and apply your formatting.

Typing text: When you're ready to get started, type in the document area of WordPad. A flashing vertical line called the cursor | indicates where the next text that you type will appear. To move the cursor within text, click where you want the cursor to appear.

Unlike using a typewriter, when you type in WordPad you don't have to press ENTER to start a new line. WordPad will take care of that for you by automatically starting a new line when you reach the end of the one you're working on. When you want to start a new paragraph, press ENTER.

Selecting text

Whenever you want to do something to text in your document, such as copy it or format it, you need to select it first. To select text, position the mouse pointer to the left of where you want to begin your selection. With the pointer positioned, click and hold the left mouse button while you drag the pointer over the text that you want to select. The selected text will be highlighted. Once you've completed your selection, release the mouse button.

Copying and moving text

If you have text in one part of your document that you want to appear in another part, there are two ways to move the text around without having to retype it: You can copy and paste it to another location, or you can move it to another location.

When you copy text, it is placed in the Clipboard. You can then paste it in a different location. The original text is preserved.

When you move text to another location, the original text is not preserved. You might find this method useful when you want to rearrange the sentences and paragraphs in your document.

To copy text and paste it in another location

1. Select the text that you want to copy.
2. On the **Edit** menu, click **Copy**.
3. Move the cursor to the location where you want to insert the copied text.
4. On the **Edit** menu, click **Paste**.

Tip

- To quickly copy text that you've selected, press CTRL+C. To paste it, press CTRL+V.

Inserting and deleting text

WordPad makes it easy to insert and delete text wherever you want. To insert text, click where you want to insert the text and then just start typing. To delete text, select the text that you want to delete, and then press DELETE.

Formatting your document

Formatting refers to the how the text in your document looks as well as how it is arranged. WordPad lets you easily change the formatting in your document. For example, you can choose from many different fonts and font sizes, and you can make your text almost any color you want. You can also easily change how your document is aligned.

To change the font, font style, or font size:

1. Select the text whose formatting you want to change.
2. On the **Format** menu, click **Font**.
3. In the **Font** box, type or select the font you want to use.
4. In the **Font Style** box, type or select the font style you want.
5. In the **Size** box, type or select the size you want to use.

To change the color

1. Select the text whose color you want to change.
2. On the **Format** menu, click **Font**.
3. Under **Color**, click the color that you want.

To change the alignment

You can align the text in your document to either the left margin, the center, or the right margin.

1. Select the paragraph whose alignment you want to change.
2. On the **Format** menu, click **Paragraph**.
3. In the **Alignment** box, click the alignment you want.

Saving your document

It's a good idea to save your document periodically while you are working on it so that in case your computer stops working for some reason, you won't lose any of your work. Saving the document will also allow you to come back to it later if you want to work on it again.

To save the document

- On the **File** menu, click **Save**.

If you haven't saved the document yet, you'll be asked to provide a name for the document and location on your computer to save it to:

1. In the **Save in** box, click the location where you want to save the document.
2. In the **File name** box, type a name for your document.
3. Click **Save**.

Printing your document

On the **File** menu, click **Print**. In the **Print** dialog box, use the **Page Range** box and the **Number of copies** box to specify which pages you want to print as well as how many copies. When you're done, click **Print**.

MS-OFFICE:

Microsoft Office is an office suite of desktop applications, servers and services for the Microsoft Windows and OS X operating systems, introduced by Microsoft on August 1, 1989. Initially a marketing term for a bundled set of applications, the first version of Office contained Microsoft Word, Microsoft Excel, and Microsoft PowerPoint. Over the years, Office applications have grown substantially closer with shared features such as a common spell checker, OLE data integration and Microsoft Visual Basic for Applications

scripting language. Microsoft also positions Office as a development platform for line-of-business software under the Office Business Applications brand. Office is reported to now be used by over a billion people worldwide.

The current versions are Office 2013 for Windows, released on October 11, 2012 and Office 2011 for OS X, released October 26, 2010. On 24 October 2012, the RTM final code of Office 2013 Professional Plus has been released to TechNet and MSDN subscribers for download. On 15 November 2012, the 60-day trial version of Office 2013 Professional Plus was released for download.

MS- WORD

Microsoft Word is an example of a program called a "word processor." The key benefit to using a word processor is that you can make changes easily, including correcting spelling; adding, deleting, formatting, and relocating text; and inserting images. Once you create a document, you can effortlessly print it (as many copies as you want!), save it for later modifications, or send it to a friend via e-mail. Microsoft Word is a very powerful word processor—this handout was created using just a small number of its features!

Microsoft Word is available on both PCs and Macs, so what you learn in class today should be applicable to any computer you use. The program may look slightly different depending on the version and computer that you're using, but Microsoft Word will function in the same basic ways. There are other word processors out there, including OpenOffice.org Writer, WordPerfect, Apple Pages, and WordPad. They have many features in common with Microsoft Word, and you should feel free to choose any program you prefer.

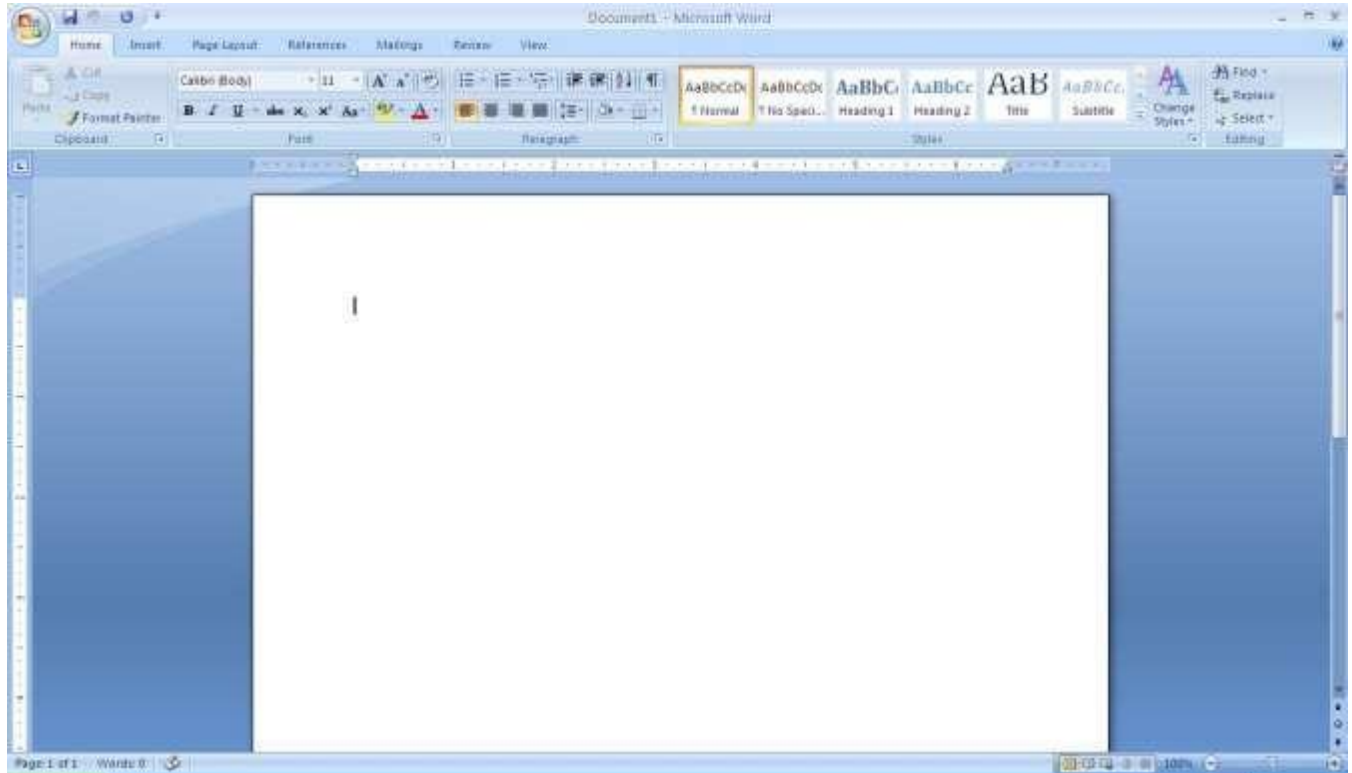
Opening Microsoft Word

To get started with Microsoft Word (sometimes called "MS Word"), you will need to locate and open the program on your computer. To open the program, point to Word's icon on the desktop with your mouse and double-click on it with the left mouse button. If you don't see the MS Word icon on your desktop, you'll have to access the program from the Start Menu. Click on the button in the bottom left corner to pull up the Start Menu. You should see the MS Word icon here, so click on it once with your left button. If you still don't see it, click on "All Programs" and scroll through the list of programs until you find it. Click once with your left button to open the program. Occasionally, Microsoft Word will be in a folder called "Microsoft Office" or something similar—this will make one more step between "All Programs" and "Microsoft Word."

MS Word will then open a blank page called "Document 1."

This is an image of the upper-left corner of MS Word. This box features two important pieces of information: the name of the file that you are currently working on (in this case, "Document 1" since we have not yet renamed it) and which program you are using ("Microsoft Word"). There should now be a blinking line in the top left corner of the part of the screen that looks like a sheet of paper. This means you will be able to type in this space!

Microsoft Word Features



The Title Bar

This is a close-up view of the Title Bar, where file information is located. It shows the name of the file (here, "Microsoft Word") and the name of the program ("Microsoft →ord"). Notice the three buttons on the right side, controlling the size and closing of the program.

Minimize: Left click this button to shrink the window down to a small button that will appear in the task bar

Maximize: Left click this button to make the window as large as it can be—it should take up your entire screen.

Close: Left-click this button to close the window. The program will close and stop running. Make sure you save your work first!

Restore Down:

Left-click this button to make the window smaller without minimizing it. The tabbed Ribbon menu system is how you navigate through Word and access various Word commands. If you have used previous versions of Word, the Ribbon system replaces the traditional menus.

The File Menu

In Microsoft Office 2007, there was something called the Microsoft Office Button () in the top left-hand corner. In Microsoft Office 2010, this has been replaced with a tab in the Ribbon called "File."

When you left-click on this tab, a drop-down menu appears. From this menu, you can perform the same functions as were found under the Microsoft Office Button menu, such as: Create a new document, open files, save files in a variety of ways, and print.

Quick Access Toolbar

On the top left-hand side of the Title Bar, you will see several little icons above the File menu. These let you perform common tasks, such as saving and undoing, without having to find them in a menu. →e'll go over the meanings of the icons a little later.

The Home Tab

The most commonly used commands in MS Word are also the most accessible. Some of these commands available in the Home Tab are: The Home Tab Toolbar offers options that can change the font, size, color, alignment, organization and style of the text in the document. For example, the "Times New Roman" Indicates the FONT of your text, the "12" indicates the SI↑E of your text; etc. →e will go over how to use all of these options toformat your text in a little while. Each of these options expands into a menu if you left-clickon the tiny down-arrow in the bottom right corner of the window. The ruler is found below the Ribbon. The ruler tells you where you are on the page, along with the dimensions ofthe overall document. Just like a real piece of paper, the default setting is 8.5 x 11 inches,and the margins have been incorporated for you. Similar to most options in MS Word, this is entirely customizable and the user can create a document of any dimensions.

Keyboard Review

In order to use MS Word effectively, you must input commands using both the mouse and the keyboard. The above image of a keyboard should look similar to the keyboard in front of you; learning just a few certain keys will help to improve your efficiency in typing, as well as present you with more options within the program. The following is a list of commonly used keys that have special functions in MS Word (key functions can change depending on which program you are using) :

1. Backspace: This key deletes letters backward.
2. Delete: This key deletes letters forward.
3. Shift: This key, when pressed WITH another key, will perform a secondary Function.
4. Spacebar: This key enters a space between words or letters.
5. Tab: This key will indent what you type, or move the text to the right. The default Indent distance is usually ½ inch.
6. Caps Lock: Pressing this key will make every letter you type capitalized.
7. Control (Ctrl): This key, when pressed WITH another key, performs a shortcut.

8. Enter: This key either gives you a new line, or executes a command (pressed in MS Word, it begins a new line).
9. Number Keypad: These are exactly the same as the numbers at the top of the Keyboard; some people just find them easier to use in this position.
10. Arrow Keys: Like the mouse, these keys are used to navigate through a document.

MICROSOFT WORD BASICS

Typing Text

To enter text, type just as you would if you were using a typewriter. To capitalize a letter, hold down the SHIFT key while typing the letter. Or, press the CAPS LOCK key on the left-hand side of your keyboard. You will have to press the CAPS LOCK key again—once you are done capitalizing—to remove the lock. You do not need to press ENTER to start a new line—Microsoft Word will automatically wrap your sentence at the end of the line. To move the cursor from its position at the end of your sentence to anywhere else on the page, use the mouse or the arrow keys to move the cursor where you want the letters or spaces to be (left-click the mouse to place the cursor) and then type what you want to add—the text will automatically adjust to include it. Press ENTER to start a new paragraph (this is sometimes called a “carriage return”).

Deleting Text

While typing a document, sometimes you will make a mistake. Unlike a typewriter, MS Word can delete text on the screen and leave no trace—it is as if you never typed on the page in the first place. There are two different buttons on the keyboard that will allow you to erase text. They are the “Delete” key and the “Backspace” key (#s 1 and 2 on the keyboard map on page. The “Backspace” key deletes words to the LEFT of the cursor, and the “Delete” key deletes words to the RIGHT of the cursor.

BACKSPACE ↔ DELETE

Let’s assume that the vertical line dissecting the word “creative” in the example above is

Our cursor. Pressing the “Backspace” key will erase “a,” “e,” “r,” “c,” and so on, moving LEFT.

The “Delete” key will erase “t,” “i,” “v,” “e,” and so on, moving RIGHT.

To delete a whole chunk of text at once, left-click with your mouse and drag to highlight a section of text. Then simply press "Backspace" or "Delete" and all of the highlighted Text will disappear.

Undoing and Redoing

The UNDO and REDO features of Microsoft Windows applications are great tools to rely on. The program will keep a list of the last 25 commands that you have performed, and it allows for taking "one step" backwards in order to erase what you have just done. Click on the UNDO button in the Quick Access Toolbar to go back one step. Click on the REDO button in the Quick Access Toolbar to go forward one step.

Formatting Text

Changing the look of what you've written is called "formatting." This can include changing the text style, size, color, and more. This is a sentence that features many different fonts. This is a sentence that features many different sizes. From the Home Tab, the "B" will make your text BOLD, the "I" will put it in italics, and the "←" will add an underline to your text. These features do not have to be used independently of each other—in other words, you can bold, underline, and italicize a single piece of text. The alignment of the text can be altered using the buttons with the horizontal lines on them. You can align text To the LEFT, To the CENTER and to the RIGHT.

In order to apply certain stylistic or formatting changes to text, you must first HIGHLIGHT the text. This is a common procedure in Microsoft Windows applications, And because it is so useful, it is a skill worth practicing. You can also change the font color by clicking on the appropriate buttons in the

Formatting Toolbar. Experiment and remember—you can always start over with a fresh, new document, so don't worry about making mistakes! If you do, however, there is always the incredible "←NDO" tool. Cutting, Copying, and Pasting Text in MS Word, you can CUT or COPY text from one area of the document and save that text to be PASTED elsewhere (these commands are found on the Home Tab). When you CUT text, you actually delete it from where you took it, as opposed to COPYING it, which makes a copy of your selection.

When you CUT or COPY text, it is stored on the CLIPBOARD. The Clipboard is a tool in MS Word that stores cuts and copies of your work, to be pasted in other places in the document. Once your selection is on the CLIPBOARD, you can PASTE it as many times as you want! CUTTING a selection will place it on the clipboard, just in case you want to PASTE it elsewhere. To CUT a selection, first highlight it. Then, click on the CUT icon from the Home Tab. The highlighted text will disappear, as you have just cut it out, but a copy is now on your clipboard, and MS Word is waiting for you to paste it somewhere else.

To PASTE this cut selection, place your cursor where you want the selection to go. Click on the PASTE icon from the Home Tab toolbar, and it will pop the text into place, right where you have your cursor. To COPY, simply follow those same steps, replacing the CUT command with COPY. The COPY command will not alter your original selection at all, as it simply makes a copy of the selection without changing or deleting it in any way.

DOING MORE WITH MICROSOFT WORD

Inserting Images and Clip Art

With Microsoft Word, you can insert pictures in your document using the Insert Tab toolbar. You can insert pictures from the "Clip Art" album that comes with the program, or you can insert pictures from a file that you have previously saved on a disk or elsewhere on the computer (e.g., the My Pictures folder). Clip Art is a collection of cartoon and computer-generated images that cover a broad array of commonly needed icons and pictures. These include business, holiday, nature, entertainment, academic and other themes, along with standard bullets and symbols.

To insert a clip art picture into a Microsoft Word document, you will first need to place your cursor (left-click) where you wish to insert your picture in your document. Then, from the Insert tab on the Ribbon, click on "**Clip Art.**" A Clip Art search box will appear on the right side of your screen. From here, you can enter a search word for the picture you would like to find. Then click "Go." This will show all the options of pictures you can insert. When you see a piece of clip art that you like, click on it, and it will automatically insert into your document, in the exact place where you left the cursor. To resize your image, grab one of the corners of the image by left-clicking and holding—drag it to your desired size and fine-tune the placement. The cursor will change from a typing icon to a double arrow when it can be used to resize an image.

Drag the corner toward the upper-left corner of the screen, and the image will grow. Drag toward the bottom-right, and it will shrink. You can move the image to another place in the document by left-clicking and holding on the center of it; drag it to its desired location. More options, such as aligning the image with text (or special effects like wrapping text around or through an image) can be found by clicking on the picture, and then choosing the Picture Tools Tab. You can also insert a photograph or other images from your files using this same process. On the Insert Tab, click on Picture and locate the files that you want to include.

Modifying Line Spacing

Line spacing in Word refers to the amount of space between lines of text. The default in Word 2010 is 1.15 spacing, which leaves a little bit more space than single-spacing, or what you would find in a normal book. Single spacing is generally easy for the eye to read. There may be times, however, when you want to change this spacing. One common option is to double-space text: This text is double-spaced. Double-spacing is especially useful if

someone else is proof reading your document. It allows for more room to write comments on the page.

To change the line spacing:

1. Select text you want to format by highlighting it.
2. On the Home Tab, click on the Line Spacing button in the Paragraph group.
3. Choose the spacing you want from the menu that appears. For more options, select Line Spacing Options. In the dialog box that appears, you can choose other spacing options, including spacing between paragraphs. This can be done by changing the values in the Before and After boxes.

Creating Bulleted and Numbered Lists

Word allows you to create lists within your document that can be organized with bullets or numbers. Lists are useful for presenting text that wouldn't make the most sense in paragraph form (for example, step-by-step instructions) or for emphasizing key points. Bullets are usually small circles at the beginning of item in a list, and numbers are used for lists that are arranged in sequential order. Here's an example of a bulleted list:

- Bananas
- Milk
- Eggs
- Ice Cream

To create a list:

1. On the Home tab, select either the Bullets or Numbering buttons from the Paragraph group. If you want to choose a particular style for your bullets or numbers, click on the triangle next to the button and choose a style from the menu that appears.
2. You will see the first bullet or number appear on your document. Type your first line of text and then hit Enter.
3. Another bullet or number will appear automatically. Type your next line of text and hit Enter.
4. When you have finished your list, hit Enter twice to end the bullets or numbering.

Changing Views and Overall Look

There will be times when you need to create documents that don't fit on a standard piece of paper (8.5" by 11").

Click on the Page Layout Tab.

From this menu, you can alter the margins, width, height, and orientation (portrait = 11"14long; landscape = 11" wide) of the document, as well as choose from a variety of templates, different types of paper, and document styles that come with the program. To change the orientation of your page, click on the Orientation button.

Click on Landscape from the drop-down box that appears. Your document should now be in landscape position. If you wish to return to portrait orientation, go through the same steps, clicking on the Landscape option instead of Portrait. If you would like to change the display of your document or zoom in and out without affecting the way your document will print, you can do that from the View Tab. Anything you change from the View Tab will not change the way your document prints out. So if you want your text to be bigger when you print out the document, make sure to change the size of the font instead of the zoom.

Spelling and Grammar Check

One benefit of using a computerized word processor is its ability to recognize, change, and give advice about your writing. MS Word has utilities that can check your spelling and grammar against a master database, and can offer advice on a variety of different grammatical styles. **MS Word** automatically underlines any words that it does not recognize in red, assuming that they are not spelled correctly, and underlines in green if it does not recognize the grammatical pattern, assuming that the sentence does not make sense.

For example: In this case, "jumping" should read "jump" and "laziy" should be "lazy." If you right click on the word, a menu will pop up with options, including "Ignore" if you do not want help on this specific phrase or agree with the suggestion. You can also choose to add words to the MS Word spelling database, if you are going to use them often and don't want them to be flagged as misspelled every time—this is especially useful for names, as MS Word often interprets these as misspellings. The spelling and grammar tools can be found in the Review Tab

USING MICROSOFT WORD

Microsoft Word is an example of a program called a "word processor." Word processors are used to create and print text documents in much the same way that you would use a typewriter. The key benefit to using a word processor is that you can make changes easily, including correcting spelling; adding, deleting, formatting, and relocating text; and inserting images. Once you create a document, you can effortlessly print it (as many copies as you want!), save it for later modifications, or send it to a friend via e-mail. Microsoft Word is a very powerful word processor—This handout was created using just a small number of its features! Microsoft Word is available on both PCs and Macs, so what you learn in class today should be applicable to any computer you use. The program may look slightly different depending on the version and computer that you're using, but Microsoft Word will function in the same basic ways. There are other word processors out there, including OpenOffice.org Writer, WordPerfect, Apple Pages, and WordPad. They have many

features in common with Microsoft Word, and you should feel free to choose any program you prefer.

Opening Microsoft Word

To get started with Microsoft Word (sometimes called "MS Word"), you will need to locate and open the program on your computer. To open the program, point to Word's icon on the desktop with your mouse and double-click on it with the left mouse button. If you don't see the MS Word icon on your desktop, you'll have to access the program from the Start Menu. Click on the button in the bottom left corner to pull up the Start Menu. You should see the MS Word icon here, so click on it once with your left button. If you still don't see it, click on "All Programs" and scroll through the list of programs until you find it. Click once with your left button to open the program.

Occasionally, Microsoft Word will be in a folder called "Microsoft Office" or something similar—this will make one more step between "All Programs" and "Microsoft Word." MS Word will then open a blank page called "Document 1." This is an image of the upper-left corner of MS Word. This box features two important pieces of information: the name of the file that you are currently working on (in this case, "Document 1" since we have not yet renamed it) and which program you are using ("Microsoft Word"). There should now be a blinking line in the top left corner of the part of the screen that looks like a sheet of paper. This means you will be able to type in this space!

MICROSOFT WORD BASICS

Typing Text

To enter text, type just as you would if you were using a typewriter. To capitalize a letter, hold down the SHIFT key while typing the letter. Or, press the CAPS LOCK key on the left-hand side of your keyboard. You will have to press the CAPS LOCK key again—once you are done capitalizing—to remove the lock. You do not need to press ENTER to start a new line—Microsoft Word will automatically wrap your sentence at the end of the line. To move the cursor from its position at the end of your sentence to anywhere else on the page, use the mouse or the arrow keys to move the cursor where you want the letters or spaces to be (left-click the mouse to place the cursor) and then type what you want to add—the text will automatically adjust to include it. Press ENTER to start a new paragraph (this is sometimes called a "carriage return").

Deleting Text

While typing a document, sometimes you will make a mistake. Unlike a typewriter, MS Word can delete text on the screen and leave no trace—it is as if you never typed on the page in the first place. There are two different buttons on the keyboard that will allow you to erase text. They are the "Delete" key and the "Backspace" key (#s 1 and 2 on the keyboard map on page).

The "Backspace" key deletes words to the LEFT of the cursor, and the "Delete" key deletes words to the RIGHT of the cursor.

Microsoft Word Features

The Title Bar

This is a close-up view of the Title Bar, where file information is located. It shows the name of the file (here, "Microsoft Word") and the name of the program ("Microsoft Word"). Notice the three buttons on the right side, controlling the size and closing of the program.

Minimize: Left click this button to shrink the window down to a small button that will appear in the task bar

Maximize: Left click this button to make the window as large as it can be—it should take up your entire screen.

Close: Left-click this button to close the window. The program will close and stop running. Make sure you save your work first!

Restore Down: Left-click this button to make the window smaller without minimizing it.

The File Menu

In Microsoft Office 2007, there was something called the Microsoft Office Button () in the top left-hand corner. In Microsoft Office 2010, this has been replaced with a tab in the Ribbon called "File." When you left-click on this tab, a drop-down menu appears. From this menu, you can perform the same functions as were found under the Microsoft Office Button menu, such as: Create a new document, open existing files, save files in a variety of ways, and print.

The Home Tab

The most commonly used commands in MS Word are also the most accessible. Some of these commands available in the Home Tab are: The Home Tab Toolbar offers options that can change the font, size, color, alignment, organization and style of the text in the document. For example, the "Times New Roman" indicates the FONT of your text, the "12" indicates the SIZE of your text; etc. We will go over how to use all of these options to format your text in a little while. Each of these options expands into a menu if you left-click on the tiny down-arrow in the bottom right corner of the window.

The Ruler

The ruler is found below the Ribbon. The ruler tells you where you are on the page, along with the dimensions of the overall document. Just like a real piece of paper, the default setting is 8.5 x 11 inches, and the margins have been incorporated for you. Similar to most

options in MS Word, this is entirely customizable and the user can create a document of any dimensions.

Keyboard Review

In order to use MS Word effectively, you must input commands using both the mouse and the keyboard. The above image of a keyboard should look similar to the keyboard in front of you; learning just a few certain keys will help to improve your efficiency in typing, as well as present you with more options within the program. The following is a list of commonly used keys that have special functions in MS Word (key functions can change depending on which program you are using)

1. Backspace: This key deletes letters backward.
2. Delete: This key deletes letters forward.
3. Shift: This key, when pressed WITH another key, will perform a secondary function.
4. Spacebar: This key enters a space between words or letters.
5. Tab: This key will indent what you type, or move the text to the right. The default indent distance is usually ½ inch.
6. Caps Lock: Pressing this key will make every letter you type capitalized.
7. Control (Ctrl): This key, when pressed WITH another key, performs a shortcut.
8. Enter: This key either gives you a new line, or executes a command (pressed in MS Word, it begins a new line).
9. Number Keypad: These are exactly the same as the numbers at the top of the keyboard; some people just find them easier to use in this position.
10. Arrow Keys: Like the mouse, these keys are used to navigate through a document or page.

MICROSOFT EXCEL

Microsoft Excel is an example of a program called a "spreadsheet." Spreadsheets are used to organize real world data, such as a check register or a rolodex. Data can be numerical or alphanumeric (involving letters or numbers). The key benefit to using a spreadsheet program is that you can make changes easily, including correcting spelling or values, adding, deleting, formatting, and relocating data. You can also program the spreadsheet to perform certain functions automatically (such as addition and subtraction), and a spreadsheet can hold almost limitless amounts of data—a whole filing cabinet's worth of information can be included in a single spreadsheet. Once you create a

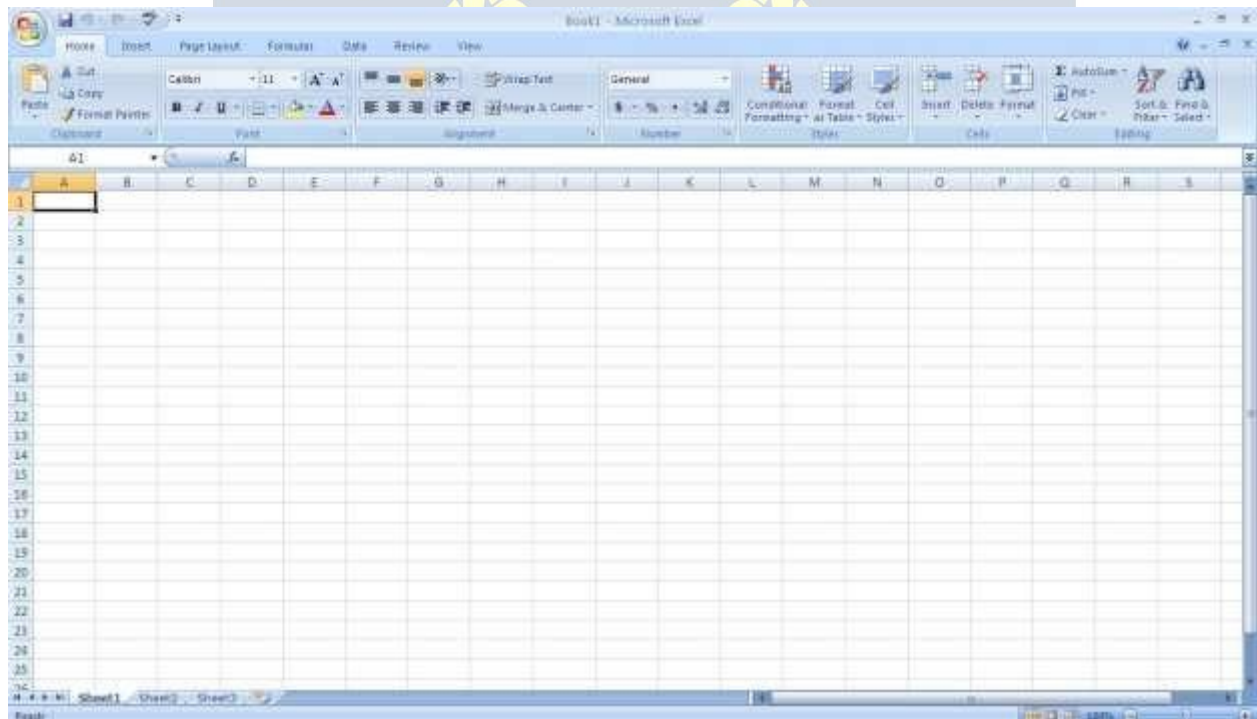
spreadsheet, you can effortlessly print it (as many copies as you want!), save it for later modifications, or send it to a colleague via e-mail. Microsoft Excel is a very powerful.

Opening Microsoft Excel

To get started with Microsoft Excel (often called "Excel"), you will need to locate and open the program on your computer. To open the program, point to Excel's icon on the desktop with your mouse and double-click on it with the left mouse button. If you don't see the Excel icon on your desktop, you'll have to access the program from the Start

Menu. Click on the button in the bottom left corner to pull up the Start Menu. You may see the Excel icon here, so click on it once with your left button. If you still don't see it, click on "All Programs" and scroll through the list of programs until you find it. It may also be located in a folder called "Microsoft Office" or something similar—it will depend on your specific machine. Click once with your left button to open the program. Excel will then open a blank page called "Book1."

Microsoft Excel Features



The Title Bar

This is a close-up view of the Title Bar, where file information is located. It shows the name of the file (here, "Book1," the default title) and the name of the program ("Microsoft Excel"). You will be able to name your file something new the first time that you save it. Notice the three buttons on the right side of the Title Bar, controlling the size and closing of the program.

The Ribbon Menu System

The tabbed Ribbon menu system is how you navigate through Excel and access various Excel commands. If you have used previous versions of Excel, the Ribbon system replaces the traditional menus. At the bottom, left area of the spreadsheet, you will find worksheet tabs. By default, three worksheet tabs appear each time you create a new workbook. On the bottom, right area of the spreadsheet you will find page view commands, the zoom tool, and the horizontal scrolling bar.

The File Menu

In Microsoft Office 2007, there was something called the Microsoft Office Button () in the top left-hand corner. In Microsoft Office 2010, this has been replaced with a tab in the Ribbon called "File." →hen you left-click on this tab, a drop-down menu appears.

From this menu, you can perform the same functions as were found under the Microsoft Office Button menu, such as: Create a new worksheet, open existing files, save files in a variety of ways, and print.6

Quick Access Toolbar On the top left-hand side of the Title Bar, you will see several little icons above the File menu. These let you perform common tasks, such as saving and undoing, without having to find them in a menu. →e'll go over the meanings of the icons a little later.

The Home Tab

The most commonly used commands in Excel are also the most accessible. Some of these commands available in the Home Tab are: The Home Tab Toolbar offers options that can change the font, size, color, alignment, organization and style of the text in the spreadsheet and individual cells. For example, the "Calibri" indicates the FONT of your text, the "11" indicates the SIZE of your text; etc. We will go over how to use all of these options to format your text in a little while. Each of these options expands into a menu if you left-click on the tiny down-arrow in the bottom right corner of the window. This tab works the exact same way as the MS Word Formatting Toolbar. The main difference is that the format changes will only affect the selected cell or cells, all unselected cells remain in the default setting ("Calibri" font, size "11").

Equation Editor

The Equation Editor is generally found below the ribbon menu. The left side denotes which cell is selected ("C5") and the right side allows you to input equations or text into the selected cell.

There are two ways to input information into a cell. You may either select an individual cell and type the equation or text into the equation editor or type the equation or text directly into the selected cell. Equations (for example, =SUM(D5+E5)) will automatically be hidden inside the cell and can only be viewed using the equation editor; the result of the equation will display in the cell. If any written text is longer than the cell width, then the spreadsheet will cover up any portion longer than the cell width. The information will still be in the cell, you just won't be able to see it at all times.

Keyboard Review

In order to use Excel effectively, you must input commands using both the mouse and the keyboard. The above image of a keyboard should look similar to the keyboard in front of you; learning just a few certain keys will help to improve your efficiency in typing as well as present you with more options within the program. The following is a list of commonly used keys that you may already be familiar with:

1. Backspace: This key deletes letters backwards.
2. Delete: This key deletes letters forward.
3. Shift: This key, when pressed WITH another key, will perform a secondary function.
4. Spacebar: This key enters a space between words or letters.
5. Tab: This key will indent what you type, or move the text to the right. The default indent distance is usually ½ inch.
6. Caps Lock: Pressing this key will make every letter you type capitalized.
7. Control (Ctrl): This key, when pressed WITH another key, performs a shortcut.
8. Enter: This key either gives you a new line, or executes a command.
9. Number Keypad: These are exactly the same as the numbers at the top of the keyboard; some people just find them easier to use in this position.
10. Arrow Keys: Like the mouse, these keys are used to navigate through a document or page.

Pointer Shapes

As with other Microsoft programs, the pointer often changes its shape as you work in Excel. Each pointer shape indicates a different mode of operation. This table shows the various pointer shapes you may see while working in Excel.

MICROSOFT EXCEL BASICS

Formatting Cells

Cells are the small rectangular boxes that make up the spreadsheet. All the information entered into an Excel spreadsheet is entered into cells. The cell width and height will usually need to be adjusted to view all the information entered into a cell.

To adjust the cell width, move the mouse pointer in between two cell columns in the column header. Hold down the left mouse button and drag the mouse left to shorten the width or right to expand the width. Notice that all cells within the column are automatically adjusted. Adjust the cell height using the same method. Move the mouse cursor between two rows, hold down the left mouse button and move the mouse up to decrease the height and down to increase the height. Before you begin entering data into a spreadsheet, you may already know the width and height you want your cells to have. In this case, you can adjust all the widths and heights by doing the following: Select the "square" between Column A and Row 1.

This will select ALL the cells in the spreadsheet. From the "Home" tab of the Ribbon Menu, within the "Cells" box, click on "Format," and select Row Height. You will now be asked to enter a numerical value for height. The default value is 15, but you can enter your own height value (10, 20, 25, etc.).

Repeat the same steps for Column width. From the "Home" tab of the Ribbon Menu, within the "Cells" box, click on "Format," and select Column → width. Note that the default value for the width is 8.43. Enter your own width value (5, 10, 15, 20, etc.).

For any given cell or selected cells, you can also format the way your data is represented within the cell(s). Select a single cell or multiple cells.

Again, from the "Home" tab of the Ribbon Menu, within the "Cells" box, click on Format." Select "Format Cells." The format window will now appear, giving you a wide variety of options on how to format your cell.

Number – This allows you to choose how to represent the numbers that are entered into a cell (number, currency, time, etc.).

Alignment – This determines how the data will be aligned within the cell (left-side, centered, or right-side).

Font – Select the type of font to be used within the cells.

Border – This option lets you choose what type of border, if any, you would like around the cells or some of the cells.

Fill – This allows you to change the background color of the cell.

Protection – This option allows you to "lock" cell information so that other users cannot make changes. **Typing in Cells**

Click on a cell to begin typing in it. It is that easy! When you are finished typing in the cell, press the Enter key and you will be taken to the next cell down. You can then begin typing in that cell. You can easily navigate around the cells using your arrow keys. Keep in mind that the Formatting toolbar in Microsoft Excel 2010 is exactly the same as the one used for Microsoft Word 2010. The biggest difference between the two programs is that, in Excel, the format is set for each individual cell.

So if you change the font and applied the bold option in cell C5, then this format will only be applied to cell C5. All remaining cells will remain in default mode until they have been changed. Sometimes you may only wish to adjust the format of one particular cell. In this case, simply select the cell by clicking the mouse on it and make any necessary adjustments to the font, size, style, and alignment. Those changes will not carry over when you begin typing in a new cell. Other times, you may wish to adjust the text format of a group of cells, entire rows, or entire columns.

In Excel, you can choose groups of cells in rectangular units—all the cells you select must form a rectangle of some kind. To select a group of cells, begin by clicking on the cell that would be in the upper-left hand corner of your rectangle. Hold down the Shift key on your keyboard and use the arrows (, , ,) on the keyboard to expand the selection of cells, or click and drag your mouse. Once the group of cells has been selected, you can make adjustments to the font, size, style, and alignment and they will be applied to all selected cells.

To select an entire row, click on the Row Number with your mouse—note how the entire row becomes highlighted. All formatting changes will now be applied to the whole row. To select an entire column, click on the Column Number with your mouse—again, the entire

column will become highlighted. All formatting changes will be applied to the whole column.

Inserting Rows and Columns.

When you are working on a spreadsheet, you may realize that you left out a row or column of data and need to add it in. To insert a row, click on the row below where you want your new row to be (remember to click on the row number to highlight the entire row). From the "Home" tab, within the "Cells" box, click "Insert." Select

"Insert Sheet Rows." A new row will automatically be inserted and the row numbers automatically adjusted. To insert a column, click on the column to the right of where you want your new column to be (remember to click on the column letter to highlight the entire column). From the "Home" tab, within the "Cells" box, click "Insert." Select "Insert sheet Columns." A new column will automatically be inserted and the column letters automatically adjusted.

Sorting Data

Once you have created your spreadsheet and entered in some data, you may want to organize the data in a certain way. This could be alphabetically, numerically, or another way. Let's look at the following spreadsheet as an example. This information can be sorted by check number, date, alphabetically by description, or using any of the other columns.

First, select all the cells that represent the data to be sorted, including the header descriptions (Check No., Date, Description, etc.). Then, select the first cell in Row 1 (Check No.) Click and drag to select all the cells that you want to sort. Using the mouse, select Sort & Filter from the Editing panel. Select Custom Sort...

Select the column you wish to sort by. Do you want to sort by alphabetical order, reverse alphabetical order, date, or amount? →hen you press "OK," your spreadsheet will be sorted in the order that you specified.

AutoSum and Excel Equations

One of the most powerful features of Excel is its ability to perform basic math functions on data. Excel can add, subtract, multiply, divide, find the average, and perform general counting functions on the numerical data that you enter. To enable this feature, highlight all of the cells in a column, plus one additional empty cell in which to display the result.

Select the AutoSum icon from the ribbon menu:

If you click directly on the Σ , Excel will automatically add up the numbers you have selected. If you click on the little dropdown arrow next to it (\blacktriangledown), you will get the full choice of mathematical functions. If you double-click on the cell in which the answer appears, you will see an equation that looks something like this (you will also see this equation in the Equation Editor):

Let's break down what exactly the equation means:

= indicates that you are starting an equation in this cell.

SUM tells the function to be performed. In this case, all the cells will be added together.

() The parentheses contain the cells that the function will be performed on.

D2 This is the first cell to be included in the addition formula.

D8 This is the last cell to be included in the addition formula indicates that all cells between the first and the last should be included in the formula.

The spreadsheet will often “select” the cells that it thinks you wish to include. But you can manually change the cell range by typing into the Equation Editor. When you are ready to execute the formula, just press the “Enter” key. Other mathematical functions you can perform from the AutoSum button include:

Average – This function will calculate the average of the selected cells.

Count Numbers – This function simply counts the number of cells selected.

Max – This function will return the highest value of the selected cells.

Min – This function will return the lowest value of the selected cells.

Remember Excel equations are similar to programming languages, so have some patience and if at first you don’t succeed, try again. Even Excel professionals create incorrect formulas on their first try.

Once you get an equation to work, you will technically be a computer programmer!

Creating Charts and Graphs:

In Excel, there are also ways to represent your data in chart or graphical forms. To create a chart or graph, select the Insert tab from the Ribbon Menu bar. In the middle of this new menu, you will see a “Charts” box.

1. Select the range of data to be represented in the chart or graph. Click on your spreadsheet and select the data to be represented using the same method that you used to select data in the sorting exercise. For this example, we want to see a visual comparison of how much money we spent on each item in the list. Select rows 1 through 8 in rows B, C, and D.

2. Select the type of chart or graph you wish to create (for our example, we’ll choose a bar graph).

3. Once you have created your graph, you can now “customize” it by giving it a title and labeling different parts. You can also make certain design decisions regarding the appearance of your graph or chart by choosing the different elements under the Design tab that appears on the Ribbon Menu bar.

4. Finally, you will need to decide if your chart should be pasted on to the existing spreadsheet or if it should be pasted on to a brand new sheet. On the very right side of the Ribbon Menu bar, select Move Chart. Once the chart or graph has been created and you realize a mistake has been made or it did not turn out the way you wanted it to, simply click on the chart or graph and hit the Backspace key on your keyboard to delete it from your spreadsheet. Don’t be afraid to go back and try again!

Multiple Sheets

Sometimes you may find it useful to have multiple spreadsheets for related data. For example, personal finances are usually tracked on a monthly calendar and it’s generally good idea to keep records of the past transactions. Excel can create multiple spreadsheets attached to the same Excel file, one for each month. Tabs for the different sheets are located in the lower-left corner of the spreadsheet. The spreadsheet that is currently being worked on is on the top tab. To access another spreadsheet, select the appropriate tab,

such as Sheet 2. The default names for the sheets are usually Sheet1, Sheet2, and Sheet3. To rename a sheet or to create a new sheet, simply move the mouse pointer over the sheet name and right-click. Select "Rename" to rename the sheet or select "Insert" to create a new sheet.

You can transfer cell information between two or more sheets by creating a simple formula. For example, most people who keep track of their personal finances don't begin every month with a balance of \$0. Instead, the balance at the beginning of the month is transferred over from the previous month. So if you have a March 2012 finance spreadsheet and an April 2012 finance spreadsheet, you can select a cell in the April 2012 sheet and write in the formula

='March 2012'!D11

The equals sign (=) tells Excel that the value for this cell is to be derived from somewhere in the spreadsheet (such as a formula or another cell location).

'Sheet name' tells Excel what sheet to refer to when it retrieves the value. The single quotation marks (' ') are important. ! D11 tells Excel that it is to copy the value that is located in cell D11 (or whatever cells you wish to copy the value from).

If you want to transfer cell information from one cell to another on the same sheet, you can use the exact same formula; just exclude 'Sheet name.'

DOING MORE WITH MICROSOFT EXCEL

Cutting, Copying, and Pasting Data

When you are typing in your spreadsheet, it can be very helpful to cut or copy information from one cell and paste it into another. This can help reduce the amount of typing you have to do, as well as limit the number of typographical errors. Cutting, copying, and pasting the contents of cells in Excel is very similar to the way you do it in Microsoft Word. These commands are all accessible from the Home tab on the Ribbon Menu bar. **To CUT** the contents of a cell, click on the cell to select it, then click on the Cut button on the Home tab. You will see the border of the cell turn into little "marching ants." The contents are now on the Clipboard, ready for you to paste them somewhere else.

To PASTE the cut selection somewhere else, click on the cell where you want the selection to go. Then click on the Paste button on the Home tab. The selection will now appear in the new cell and be removed from the original cell.

To COPY the contents of a cell, simply follow the directions above but replace the CUT command with COPY. This will leave the contents of the original cell in place and make an exact copy of them in another cell.

Filling Data across Columns and Rows

Another way to copy text from one cell to another is to drag it or "fill" it across a range of cells (a row or a column). This is a great method if you want all the text (or numbers) in column or row to be the same. For example, if you have a spreadsheet for your yearly

budget with a column for rent, you could use this method to fill in all the cells in the rent column with the same number without re-typing.

To fill the contents of cells in a column:

1. Click on the cell with the information you want to copy to the other cells.
2. Hover your mouse over the little square (called a "fill handle") in the bottom right corner of the highlighted cell border. Your mouse will turn into a cross.
3. Click on the fill handle and drag your mouse (while still holding the button down) down the column until you've highlighted all the cells where you want the text to go.
4. Let go of the mouse button. The text will now appear in all the cells you highlighted.
5. You can follow these same steps to fill the contents of a row rather than a column— simply drag your mouse across the row instead of down the column.

You can also use this technique to copy formulas across columns and rows in the spreadsheet.

CLOSING MICROSOFT EXCEL

Saving Spreadsheets

When you finish your spreadsheet and want to leave the computer, it is important to save your work, even if you are printing a hard copy. To save your work in Excel, it is essential to know **WHAT** you are trying to save and **WHERE** you are trying to save it.

Click on the File Tab, then click "**Save As**" to get started. You can change the filename that Excel has chosen just by typing a new one in the "File name" box at the bottom of the window that appears. The My Documents folder on your computer's hard drive is a good place to store your documents. A blank CD or a USB jump drive are great portable storage options and can contain a LOT of data. Excel will automatically save your document with the suffix ".xlsx"—this is simply a tag that lets Excel know that your work is specific to this program and what version it is in. You do not have to type it—just highlight what is there (default is "Book1") and write a new file name. You may also choose to save it in an older format so that it can be opened with older versions of Excel. After the first save, you can just click "Save" to preserve your work.

However, it is important to note that every following command of SAVE will overwrite your original file, creating the most up-to-date version. If you would like to keep saving different versions of your worksheet, be sure to use the "Save As" function each time you save, using a slightly different name for each version.

Printing Spreadsheets

To print your Excel document, click on the File Tab, then click "Print" from the left-side column. A Print Preview and printing options will appear in your Excel window.

Click the large "Print" button to send your worksheet to the printer. You can change the number of copies you would like to print, change the paper orientation, choose which printer you want to use, and more.

Finding More Help

You can get help with Excel by clicking on the Question Mark symbol in the upper-right hand corner of the main menu bar or by pressing the "F1" button. This will take you to help from Office.com, Microsoft's help website. There are also many other resources and tutorials available online. You might try a Google search with the words "Excel 2010" and the function you are trying to perform. Ask your instructor for help finding these resources if you have any trouble.

Closing the Program

Click on the File Tab, then click "Exit."

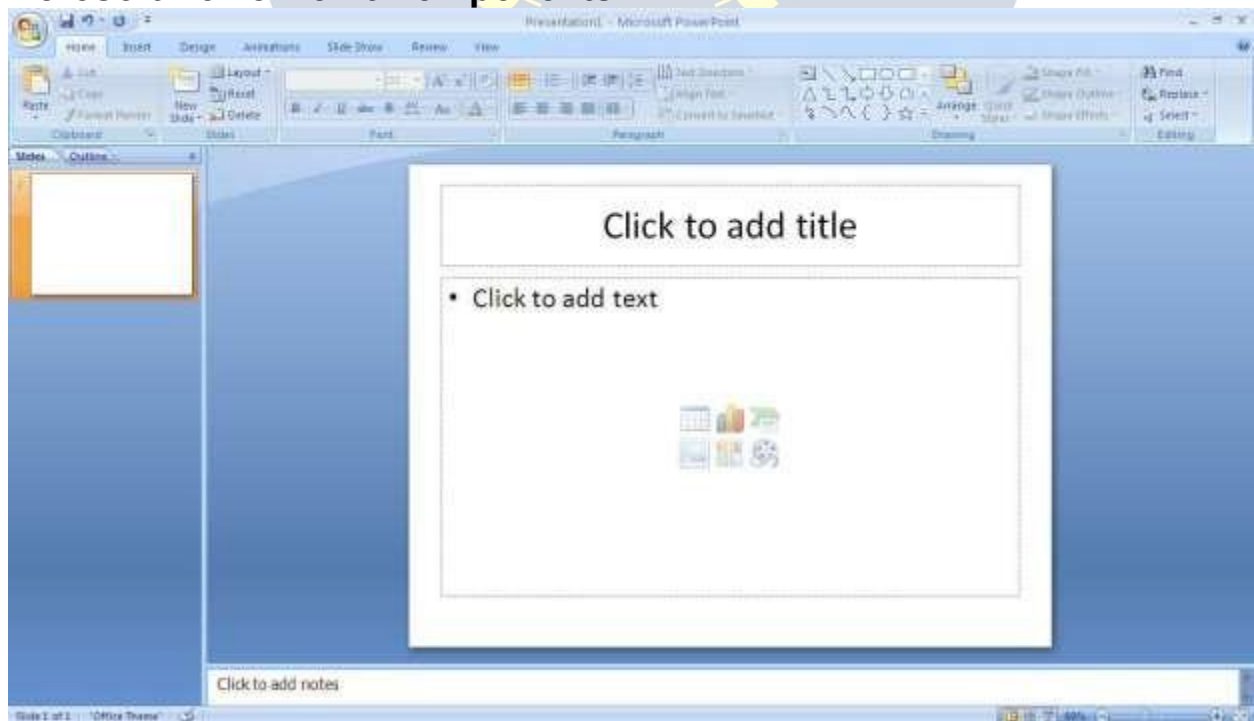
OR

Click on the X in the top right corner of the Excel screen.

MICROSOFT POWERPOINT

Microsoft PowerPoint is a professional presentation program that allows the user to create "presentation slides" that can be displayed on the computer screen or through a projector that is plugged in to the computer. A PowerPoint presentation is a good way to convey pieces of information, usually in the form of an outline, to a large audience. Generally, PowerPoint presentations are appealing to users because they are easy to create and edit and generally small enough to fit onto a CD or a USB Jump Drive. Therefore, a user does not have to carry around any slides or a slide project, and, if necessary, can make last-minute changes to the presentation.

Microsoft PowerPoint Components:



Before you get started with Microsoft PowerPoint (commonly referred to as PowerPoint), you will need to locate and open it on the computer. It may be on your desktop.

On the computer desktop:

1. Double-click on the MS PowerPoint icon

If the MS PowerPoint icon is not on the desktop, go to the Start menu:

1. Click ► Start ► Programs ► Microsoft PowerPoint*

MS PowerPoint will open a blank page called "Presentation1."

The Title Bar

This is a close-up view of the Title Bar, where file information is located. Notice the default title "Presentation1." You will get a chance to rename your presentation slides the first time you choose to save it.

The Menu Bar

The Menu Bar is a common sight in almost all Microsoft Windows programs. It features text based menus, on which virtually every option available to the user in the entire program is listed.

Each menu expands when clicked (with the left mouse button), offering many options categorized by specific tasks.

Standard Toolbar

Toolbars provide "shortcuts" to commands also found in the Menu Bar. Toolbars are usually located just below the Menu Bar and exist to offer another way to perform the same task.

The most commonly used commands in MS PowerPoint are also the most accessible.

Some of these commands are:

New | Open | Save | Print | Preview | Spell-check | Copy | Paste | Undo | Insert

Formatting Toolbar

The Formatting Toolbar offers options that can change the font, size, color, alignment, organization, and style of the text in the presentation slides. For example, (starting from the left side of the Toolbar) the "Arial" indicates the FONT of your text; the "32" indicates the SIZE of your text; and so on. This toolbar works the exact same way as the MS Word Formatting Toolbar. The main difference is that the format changes will only affect the text box in which you are currently working.



All other text boxes will remain in the default setting ("Arial" font, size "32"). Design/New Slide Icons additional shortcut icons appear on the MS PowerPoint toolbar: the Design icon and the New Slide icon. Both commands are frequently used, and it is good to be familiar with their location on the toolbar.

The Design icon is a shortcut to the slide design page. Here you will be able to create and edit the appearance of your slide(s). The New Slide icon automatically adds a new, blank slide for you to work on. You can keep track of the slides you have already worked on in the Slide Outline box on the left-hand side of the screen. You can access a slide at any time by clicking on it with your mouse through this screen.

SIMPLE TASKS IN MICROSOFT POWERPOINT

Text Boxes:

Text Boxes are "designated" areas that allow you to type words, sentences, and bullet points into the slide. You can adjust the size and placement of the text box within any given slide. It is also possible to have multiple text boxes per slide.

When you first open MS PowerPoint, there will automatically be two text boxes on the slide: "Click to add title" and "Click to add subtitle." These text boxes already have a preset format applied to them. The "title" box has a font size of "44" while the "subtitle" box has a font size of "32." You can change the text format of any box at anytime by adjusting the format settings on the Formatting Toolbar. To adjust the size of the text box, first click on the text box. Notice the change in border. Once you have clicked on a text box, the border of the box becomes thicker and little circles appear on the corners and at the midpoints of the box. Move the mouse pointer over any one of the circles. Notice that the mouse pointer will change to either  or . Click and hold down the left mouse button. To adjust the height, move the mouse up or down; to adjust the width, move the mouse left or right. Note that the corner circles can adjust both height and width, while the mid-point circles can only adjust either height or width.

To move the text box to a different location on the slide, move your mouse pointer over any part of the thick, gray box outline. Notice the change in your mouse pointer (it will look something like a "plus" sign (+) with arrows). Click and hold down the left button on your mouse. You can now drag the text box to any position on the slide by simply moving your mouse. To write in a text box, simply click inside the box with your mouse. When a cursor is flashing, you are ready to type.

Slide Design:

One of the more popular features of MS PowerPoint is that it gives the use a wide variety of design choices. At some point, you may wish to change the design and color of the background of your slides. To do so, simply click on the Design icon. On the righthand side of your screen, the "Slide Design" pane will appear. Here, you are given three design options to apply to your slides.

Design Templates: Here is a list of over 20 templates that you can choose from to apply to the background of your slides. For each template, you have the option of applying the design to all slides or to only the selected slide. You can view these options by clicking on the arrow on the right side of the template. If you are connected to the Internet, you can also choose to view hundreds of additional templates by choosing "Design Templates on Microsoft Office Online" on the bottom of the template screen.

Color Schemes: This option gives you a variety of color schemes to choose from to apply to your selected design template.

Animation Schemes: This option will allow you to select different types of animation that can be applied to either all the slides or selected slides. The animation schemes are

different ways the slides are "introduced" to your audience, such as "faded wipe" or "pinwheel."

Designing your own background:

Sometimes you may wish to design your own background.

To do this, select Format ► Background... from the Menu Bar. Select the Color Bar on the bottom of the Background window and choose the appropriate color and or/effects that you desire. There is no right or wrong way to design your own background, so if you choose this option, have some patience and don't be afraid to experiment.

Slide Layout:

The default in PowerPoint for each new slide is to have a Title text box and a Text box. As discussed earlier, you do have the option of rearranging the text boxes to fit any layout you have in mind. However, sometimes it will be easier simply to select a layout that fits your needs rather than reformatting the default layout for each slide. To choose a slide layout, select Format ► Slide Layout from the Menu Bar. On the right-hand side of your screen, the "Slide Layout" pane will appear. From here, you have the option of applying a slide layout either to a selected slide or to a new slide that has yet to be inserted. You can view these options by clicking on the arrow on the right side of the template.

Note the four different types of layout categories.

Text Layout: These options are for choosing a layout in which you are only organizing different text boxes. There is no room for pictures, charts, or graphs.

Content Layout: These options are for choosing a layout that will only contain pictures, charts, or graphs. There is no room for text boxes.

Text and Content Layout: These options have room for both text boxes and pictures, charts, or graphs.

Other Layouts: These layouts are designed for other media content such as video clips or audio tracks.

Inserting a Picture:

Inserting a picture into MS PowerPoint is very similar to inserting a picture into MS Word.

1. Select the slide that you wish to add a picture to.
2. From the Menu Bar, select Insert ► Picture ► From File (or Clip Art).
3. Choose the picture you wish to insert and select "Insert."

***Note*:** The picture will be inserted in its original size, so you will probably have to resize it. You can re-size a picture the same way you re-sized a text box. Click on the picture and note the circles at the corners and at the midpoints of the length and width. Move your mouse pointer over the circles, left-click the mouse and hold the button down. Move the mouse vertically or horizontally to re-size.

Rotating a picture: It is possible to rotate pictures in MS PowerPoint. Once a picture has been inserted onto a slide and you have clicked on the picture so it is "outlined," notice the green circle above the center of the picture.

Move the mouse pointer over the green circle, and left-click and hold the button down. Now move the mouse either clockwise or counter-clockwise and the picture will rotate in the same direction.

Inserting a Graph or Table:

There are two ways to insert a graph or table into a PowerPoint slide.

1. The graph or table may already exist in another document or file. If this is the case, then you can simply insert the graph or table as you would a picture. Follow the instructions under

Inserting a Picture.

Or you could Cut and Paste a graph or chart from another document or file into your slide presentation.

2. MS PowerPoint can create graphs or tables directly in the slide. From the Menu Bar, select

Insert ► Chart (or Table).

For the Chart option, PowerPoint will present a very simple bar graph.

You will need to go in and edit the information in the bar graph to conform to the information you wish to represent in a chart. From the Datasheet, you can enter in the appropriate values for the columns and re-name the entities.

The Table option will present you with the following window:

Select the appropriate number of rows and columns you wish to have in your table. Click **"OK"** to insert the table. Once the table has been inserted you will be able to put information into the table cells.

Slide Management:

Once you have completed a slide, you can create a new slide by clicking on the New Slide icon. Notice that your previous slides still appear on the left-side frame. You can still access your previous slides by simply clicking on them from this location.

Once all your slides have been completed, you can present your slides in a "slideshow." From the Menu Bar select Slide Show ► View Show. Notice that the slide takes over our entire screen. To navigate through the slides, use the arrow keys on your keyboard (←, →, or ↵). You can also navigate through your slides by clicking on the left or right arrows in the lower left corner of your slide. To end your slide show, click on the square box on the lower left side of your slide show (this is difficult to see at first because it is designed to blend into your slide show so it won't be noticeable to your audience). Select End Show from the pop-up window.

CLOSING MICROSOFT POWERPOINT

Saving Slide Shows:

When you come to a stopping point and want to leave the computer, it is important to save your work (even if you are printing a hard copy — saving should be a reflex). To save your work in MS PowerPoint, it is essential to know WHAT you are trying to save as well as WHERE you are trying to save it.

Click ► File ► Save from the Menu Bar to get started.

You can change the filename that PowerPoint has chosen just by typing a new one in the File name box at the bottom of the window that appears.

MS PowerPoint will automatically save your document with the suffix ".ppt" — this is simply a tag that lets PowerPoint know that your work is specific to this program. You do not have to type it — just highlight what is there (default is "Presentation1") and write a new file name.

The My Documents folder on your computer's hard drive is a good place to store your documents. A blank CD (compact disc) or a USB Jump Drive are great portable storage devices and can contain a LOT of data. It is important to note that every consequent command of SAVE will overwrite your original file, creating the most up-to-date version. If you want to save any changes to your PowerPoint slides without destroying the original one:

In the Menu Bar, click ► File ► Save As from the menu bar and give your document a new filename (unique from the original). To bring a saved document back up on the screen from MS PowerPoint:

Click ► File ► Open from the Menu Bar. Locate where the file is located (which folder, that is) and click on the filename of the document you want.

Click ► Open.

Printing Slides:

To print your MS PowerPoint slides:

Click ► File ► Print from the Menu Bar and a Print window will pop up on the screen.

Click ► OK for your document to start printing.

As with all commands in MS PowerPoint, you can make changes along the way. From the Print menu, you can alter how many copies will be made, in what order the pages will be, and much more. Other useful tools are the Print Preview function found alongside the Print command and the

Page Setup function. Print Preview will allow you to look over an exact copy of what will come out of the printer before actually executing the print command.

Closing the Program:

When you are finished,

Click ► File ► Exit from the menu bar

OR

Click on the X in the top right corner of the computer screen.

INTERNET:

VIDYAPITH ACADEMY

Introduction:

WWW.VIDYAPITH.IN

The internet in simple terms is a network of the interlinked computer networking worldwide, which is accessible to the general public. These interconnected computers work by transmitting data through a special type of packet switching which is known as the IP or the internet protocol.

Internet is such a huge network of several different interlinked networks relating to the business, government, academic, and even smaller domestic networks, therefore internet is known as the network of all the other networks. These networks enable the internet to be used for various important functions which include the several means of communications like the file transfer, the online chat and even the sharing of the documents and web sites on the WWW, or the World Wide Web.

It is always mistaken said that the internet and the World Wide Web are both the same terms, or are synonymous. Actually there is a very significant difference between the two

which has to be clear to understand both the terms. The internet and World Wide Web are both the networks yet; the internet is the network of the several different computers which are connected through the linkage of the accessories like the copper wires, the fiber optics and even the latest wireless connections. However, the World Wide Web consists of the interlinked collection of the information and documents which are taken as the resource by the general public. These are then linked by the website URLs and the hyperlinks. Therefore World Wide Web is one of the services offered by the whole complicated and huge network of the internet. The use of IP in the Internet is the integral part of the network, as they provide the services of the internet, through different layers organization through the IP data packets. There are other protocols that are the sub-classes of the IP itself, like the TCP, and the HTTP.

By the turn of the century, information, including access to the Internet, will be the basis for personal, economic, and political advancement. The popular name for the Internet is the information superhighway. Whether you want to find the latest financial news, browse through library catalogs, exchange information with colleagues, or join in a lively political debate, the Internet is the tool that will take you beyond telephones, faxes, and isolated computers to a burgeoning networked information frontier. The Internet supplements the traditional tools you use to gather information, Data Graphics, News and correspond with other people. Used skillfully, the Internet shrinks the world and brings information, expertise, and knowledge on nearly every subject imaginable straight to your computer.

What is the Internet?

The Internet links are computer networks all over the world so that users can share resources and communicate with each other. Some computers have direct access to all the facilities on the Internet such as the universities. And other computers, example privately-owned ones, have indirect links through a commercial service provider, who offers some or all of the Internet facilities. In order to be connected to Internet, you must go through service suppliers. Many options are offered with monthly rates. Depending on the option chosen, access time may vary.

The Internet is what we call a Meta network, that is, a network of networks that spans the globe. It's impossible to give an exact count of the number of networks or users that comprise the Internet, but it is easily in the thousands and millions respectively. The Internet employs a set of standardized protocols which allow for the sharing of resources among different kinds of computers that communicate with each other on the network. These standards, sometimes referred to as the Internet Protocol Suite, are the rules that developers adhere to when creating new functions for the Internet. The Internet is also what we call a distributed system; there are no central archives. Technically, no one runs the Internet. Rather, the Internet is made up of thousands of smaller networks. The Internet thrives and develops as its many users find new ways to create, display and retrieve the information that constitutes the Internet.

Who Owns the Internet?

No one actually owns the Internet, and no single person or organization controls the Internet in its entirety. The Internet is more of a concept than an actual tangible entity, and it relies on a physical infrastructure that connects networks to other networks.

Is Web and Internet the Same?

The Internet is not synonymous with World Wide Web. The Internet is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. The World Wide Web, or simply Web, is a way of accessing information over the medium of the Internet.

It is an information-sharing model that is built on top of the Internet.

WEB BROWSER:

A **web browser** (commonly referred to as a **browser**) is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI/URL) and may be a web page, image, video or other piece of content. Hyperlinks present in resources enable users easily to navigate their browsers to related resources.

Although browsers are primarily intended to use the World Wide Web, they can also be used to access information provided by web servers in private networks or files in file systems.

Uses of Internet

Internet is today one of the most important part of our daily life. There are large numbers of things that can be done using the internet and so it is very important. You can say that with the progress in the internet we are progressing in every sphere of life as it not only makes our tasks easier but also saves a lot of time. Today internet is used for different purposes depending upon the requirement. Here in this very article we have mentioned then ten best uses of the internet. Here goes the list.

WWW.VIDYAPITH.IN

- ESTD 2020 -



1) Email:

By using internet now we can communicate in a fraction of seconds with a person who is sitting in the other part of the world. Today for better communication, we can avail the facilities of Email. We can chat for hours with our loved ones. There are plenty messenger services and email services offering this service for free. With help of such services, it has become very easy to establish a kind of global friendship where you can share your thoughts, can explore other cultures of different ethnicity.

2) Information:

The biggest advantage that internet offering is information. The internet and the World Wide Web has made it easy for anyone to access information and it can be of any type, as the internet is flooded with information. The internet and the World Wide Web has made it easy for anyone to access information and it can be of any type. Any kind of information on any topic is available on the Internet.

3) Business:

World trade has seen a big boom with the help of the internet, as it has become easier for buyers and sellers to communicate and also to advertise their sites. Now a day's most of the people are using online classified sites to buy or sell or advertising their products or services. Classified sites save a lot of money and time so this is chosen as medium by most of people to advertise their products. We have many classified sites on the web like craigslist, Adsglobe.com, Kijiji etc.

4) Social Networking:

Today social networking sites have become an important part of the online community. Almost all users are members use it for personal and business purposes. It's an

awesome place to network with many entrepreneurs who come here to begin building their own personal and business brand.

5) Shopping:

In today's busy life most of us are interested to shop online. Now a day's almost anything can be bought with the use of the internet. In countries like US most of consumers prefer to shop from home. We have many shopping sites on internet like amazon.com, Dealsglobe.com etc. People also use the internet to auction goods. There are many auction sites online, where anything can be sold.

6) Entertainment:

On internet we can find all forms of entertainment from watching films to playing games online. Almost anyone can find the right kind of entertainment for themselves. When people surf the Web, there are numerous things that can be found. Music, hobbies, news and more can be found and shared on the Internet. There are numerous games that may be downloaded from the Internet for free.

7) E-Commerce:

Ecommerce is the concept used for any type of commercial maneuvering, or business deals that involves the transfer of information across the globe via internet. It has become a phenomenon associated with any kind of shopping, almost anything. It has got a real amazing and range of products from household needs, technology to entertainment.

8) Services:

Many services are now provided on the internet such as online banking, job seeking, purchasing tickets for your favorite movies, and guidance services on array of topics in the every aspect of life, and hotel reservations and bills paying. Often these services are not available off-line and can cost you more.

9) Job Search:

Internet makes life easy for both employers and job seekers as there are plenty of job sites which connect employers and job seekers.

10) Dating/Personals:

People are connecting with others though internet and finding their life partners. Internet not only helps to find the right person but also to continue the relationship.

E-mailing

Short for **electronic mail**, **e-mail** or **email** is text messages that may contain files, images, or other attachments sent through a network to a specified individual or group of individuals. The first e-mail was sent by Ray Tomlinson in 1971. By 1996, more electronic mail was being sent than postal mail.

“Email, e-mail or electronic mail is the transmission of messages (emails or email messages) over electronic networks like the internet.”

Email, sometimes written as e-mail, is simply the shortened form of “electronic mail,” a system for receiving, sending, and storing electronic messages. It has gained nearly universal popularity around the world with the spread of the Internet. In many cases, email has become the preferred method for both personal and business communication.

How It Is Used

Messages sent by electronic mail normally reach a recipient’s account within seconds. They frequently include more than just text; images and numerous types of formatted documents are now easily included as attached files. Moreover, it is no longer necessary to be sitting in front of a PC to send or receive an email. A variety of mobile devices, such as tablet computers and smart phones, make it possible manage correspondence on the go.

Most e-mail systems include a rudimentary text editor for composing messages, but many allow you to edit your messages using any editor you want. You then send the message to the recipient by specifying the recipient's address. You can also send the same message to several users at once. This is called **broadcasting**.

Sent messages are stored in electronic mailboxes until the recipient fetches them. To see if you have any mail, you may have to check your electronic mailbox periodically, although many systems alert you when mail is received. After reading your mail, you can store it in a text file, forward it to other users, or delete it. Copies of memos can be printed out on a printer if you want a paper copy. E-mail can be distributed to lists of people as well as to individuals. A shared distribution list can be managed by using an e-mail reflector. Some mailing lists allow you to subscribe by sending a request to the mailing list administrator. A mailing list that is administered automatically is called a list server.

To get a grasp of what email is it's best — the terminology indicates it — to think in equivalents of "traditional" postal mail.

- **The email message** - Instead of using a pen to write a letter on paper, you're using your keyboard to type an email message in an email program on your computer.
- **Sending the email** - When the email is finished and has been addressed to the recipient's email address, you don't put a stamp on it and post it but press the *Send* button in the email program. This makes the email message go on its journey.

- **Email transport** - Like postal services transport letters and parcel, email servers transmit email messages from sender to recipient. Usually, emails are not delivered to the recipient directly, though, but waiting at the "nearest" mail server to be picked up by them.
- **Fetching new mail** - If you've got new mail in your mailbox, you go and fetch it. Similarly, your email program can check for new email messages at your mail server and download them for you to read.

HTML/DHTML, DREAMWEAVER

Introduction to HTML

HTML (Hypertext Markup Language) is used to create document on the World Wide Web. It is simply a collection of certain keywords called 'Tags' that are helpful in writing the document to be displayed using a browser on Internet. It is a platform independent language that can be used on any platform such as Windows, Linux, Macintosh, and so on. To display a document in web it is essential to mark-up the different elements (headings, paragraphs, tables, and so on) of the document with the HTML tags. To view a mark-up document, user has to open the document in a browser. A browser understands and interprets the HTML tags, identifies the structure of the document

(which part are which) and makes decision about presentation (how the parts look) of the document. HTML also provides tags to make the document look attractive using graphics, font size and colors. User can make a link to the other document or the different section of the same document by creating Hypertext Links also known as Hyperlinks.

WEBDESIGNING

Website: Collection of Webpages and **Webpages** is made by HTML Tags.

Internet: 24 Hours presence on Internet with World Wide Reach.

Web designing: Static Site: Without Database | Photoshop (Web Templates / Homepage)

HTML | DHTML | Dreamweaver | Flash Animation | Css & Div Tag.

Web development: Dynamic Site: Database Site | Java Script | ASP.NET | PHP | Ms-Access | MySQL | Sql Server

Requirement for Website: Domain Name | Hosting (Server Space) | Website

HTML - Hyper Text Markup Language The language used to develop web pages is called HTML. HTML is the language interpreted by a Browser. Webpage are also called html documents. HTML is a set of special codes. HTML is specified on TAGS.

For creation of HTML files we required Two ways :

Editor : For writing purpose or making HTML files Note Pad is best editor it is support coding language.

Browser : For output display there are of two type of browsers Internet Explorer and Netscape Navigator.

The contents of **HTML** page is written in **Tags / Attributes**.

HTML Tags : Tags are instructions that one embedded directly into the text of the document. An HTML tag is a signal to a browser that it should do something other than just throw text up on the screen. By connection all HTML tags begin with an open angle bracket (<) and end with a Closes angle Bracket (>) **HTML** tags can be of two types :

Paired Tags : A tag is said to be paired tag if it, along with a companion tag, close the text.

Singular Tag : The second type of tag is the singular or stand-alone tag. It does not have a companion tag.

Tag Attributes & Value : Additional information supplied to an HTML tag is known as Attributes of a tag. Attributes are written immediately following the tag, separated by a space. **Like:**

<HR Size=5 Color="Red" Width=70% Align="Right">

Here HR = Tag
Size, Color, Width, Align = Attributes
5, Red, 70%, Right = Values

Basic structure of HTML Document :

Every **HTML** program has a rigid structure.

<HTML> Start Tag | Opening Script

<HEAD> Information placed in this section is essential to the inner working of

the

document and has nothing to do with the content of the document.

<TITLE> Explanation of page purpose, Display in Title Bar of I.E.

</TITLE> Closing Tag

</HEAD> Closing Tag

<BODY> Information and Links / Secondary Links / Footers means any type of Text or Graphics for Page.

</BODY> Closing Tag

</HTML> Closing Tag | Closing Script

Paragraph Breaks : A blank line always separate paragraphs in textual material. The tag that provides this functionality is **<P>**. **<P>** tag is used to create a new paragraph.

Line Breaks : When text needs to start from a new line and not continue on the same line, the **
** tag should be used.

Heading Style : HTML supports six different level of headings. The highest-level header format is

<H1> and the lowest level is **<H6>**.

Align Attribute : Alignment attributes are often used with the text, image.

Align = Left / Center / Right **<H1 Align="Left"> SoftBrain</H1>**

Drawing Lines : The tag **<HR>** draws lines and horizontal rules. The Attributes to the **<HR>** tag are :

<HR Size=5 Color="Red" Width=70% Align="Right">

Text Level Elements : Text level Elements changes the appearance to function of text.

Bold : Bold element and causes text to appear in a bold type face.

 SoftBrain

Italic : Italic element <I> and </I> causes the text to appear in Italic type face. <I>

SoftBrain </I>

Underline : Underline element <U> and </U> causes text to underline. <U>

SoftBrain </U>

Strike : Strike element indicates that text enclosed in <S> and </S> will have a line drawn in the middle of text.

<S>
SoftBrain </S>

Teletype : Teletype element changes the font like a monospaced type writer font.

<TT> SoftBrain </TT>

SuperScript : Raised one point text upper word 15th August or SoftBrainTM .

15 th August 15th August

Subscript : Text Raised down one point downward.

H₂SO₄ H2SO4

Controlling Font Size and Color : All text specified within the tags and will appear in the font size and color as specified as attributes of the Tag The attributes

:

Face = Sets the font to be specified font name.

Size = Sets the size of the text. Size can take values between 1 to 7.

Color = Sets the color of the text.

 SoftBrain Informatic Technology

Pvt. Ltd.

Bgcolor Attributes : Specifies background color using color name. <Body

Bgcolor="RED">

Text Attributes : Specifies foreground color or the color of the text.

<Body Text="Aqua">

Using Special Characters :

Non-breaking space	&nbsp;	
Copyright Sign	&copy;	Copyright ©
Registered Trademark	&Reg;	Softbrain ®
Trademark	& #8482;	Softbrain™

LIST STYLE

Lists are the best method of organizing information and presenting it in a structure fashion Basically there are three kinds of lists. They are:

Ordered List(Numbering) : An ordered list start with the tag and ends with . Each List Itemstart with tag .If tag is not present at the beginning ,the listed items appear on the same line.

Using Attributes : Type Attribute

type="1" will give Counting numbers (1,2.....)

type="A" will give Uppercase letters (A,B.....)

type="a" will give Lowercase letters (alb.....)

type="I" will give Uppercase Roman numerals.(I,II,III.....)

type="i" will give Lowercase Roman numerals.(ixia.....)

Start Attribute: The start attribute allows you to specify the beginning of the list's numbers.**(Start="1")**

Unordered List (Bullet): We use the tag and for unordered List.

Attributes of Unordered List **type="circle"** **type="square"**
type="disc"

Definition list: The definition list uses the <DL> start tag and the </DL>end tag to create a definition list. Definition list consists of two parts.

Definiton term=Appears after the Tag <DT>

Definition description=Appear after the Tag <DD>

Result In Internet Explorer : KEYBOARD

AN INPUT DEVICE

ADDITON OF GRAPHICS / IMAGES IN HTML DOCUMENTS

Other than text,HTML allows placing of static and animated images in an HTML page.

The commonly used image file types are GIF(Graphic Interchange Format) JPEG(Joint Photographic Experts Group) also called as JPG.To add an image in HTML document ,you just need to include the tag with a reference to the desired image.The (image) tag is an empty element (i.e. it doesn't require a closing tag).

Image Element Attributes:

Align Attributes with Text : Align=Top / Middle / Bottom

**Eg: **

Align Attributes with Floating Image: Align=Left / Right

Border : specifies the size of the border to place around the image.

Height : specifies the height of the image.

Width : specifies the width of the image.

Hspace : indicates the amount of space to the left and right of the image.

Vspace : indicates the amount of space to the top and bottom of the image.

Alt Attribute: ALT attribute for the <IMG..... >tag allows for substitution of a description string of text to hold the place of the image. ****

Alt working for text

`link content`

CREATING TABLES

Table present information in the form of rows and columns. To include a table in your web pages you first need to create the table structure and for that you use HTML's

Table Tags.

<TABLE> The table's beginning with a `<TABLE>` tag and the end with a `</TABLE>` tag.

<TR> The table row element `<TR>` with an optional `</TR>` end tag creates a horizontal row of cell.

<TH> The table heading `<TH>` with an optional `</TH>` creates leading cells.

<TD> The table data element `<TD>` tags with an optional `</TD>` end tag.

Table Attributes : <Align>Attribute :

`<Table Align="Left" / "Center" / "Right">`

<Border>Attribute : `<Table Border=5>`

Bgcolor Attribute: `<Table Bgcolor="Color Name Or Color Value">`

`<Tr Bgcolor="Color Name">`

`<Td Bgcolor="Color Name">`

`<Th Bgcolor="Color Name">`

Background Attribute: This sets an image on the Background of table.

`<Table Background="Computer.jpg">`

Caption : This tag is optional and it contains tables description.

`<Caption Align=Top/Bottom/Left/Right>Caption Text</Caption>` **Width & Height**

Attribute: `<Table Width="300" Height="400">` **Cell spacing Attribute:** You can use cell spacing attribute to control the spacing between each cell in the table.

`<Table Cell spacing="Number">`

Cellpadding Attribute: (Spacline between cell content and cell border)

`<Table Cellpadding="Number">`

Rowspan Attribute: The rowspan attribute is used in a cell to make the cell's content merge with another cell in a row. `<Td`

`Rowspan="Number">` `<Th Rowspan="Number">`

Colspan Attribute : Similar to ROWSPAN,COLSPAN attribute is used in a cell to make the cells content merge with another cell in a column.

`<Td Colspan="Number">` `<Th Colspan="Number">`

LINKING DOCUMENTS

LINKS: HTML allows linking to other HTML documents as well as images.

Clicking on a Section of text or an image in one webpage will open an entire webpage or an image. The text or an image that provides such linkages is called Hypertext, a Hyperlink or a Hotspot. Every Hyperlink.

1. Appears in blue color.
2. The hyperlink text / image is underlined.
3. When the mouse cursor is placed over it, the standard arrow mouse cursor changes to the shape of a hand.

The blue color, which appears by default, can be overridden. To change these link colors there are three attributes that can be specified with the `<BODY>` tag. These are:

Link: Change the default color of a hyperlink to whatever color is specified with this tag.

Alink:(Active Link)Changes the default color of a hyperlink that is activated to whatever color is specified with this tag.

Vlink: (Visit Link)Changes the default color of a hyperlink that is already visited to whatever color is specified with this tag.

The Anchor Tag: Anchor elements are used to link between two or multiple HTML documents. "Anchor" tag starts with `<A>` and ends with ``. The `<A>` tag requires the

HREF attribute and (**hyperlink reference**) attribute accepts the URL of the linked page. **content**

Every hyperlink has two components: The source and the destination.

OPEN YAHOO SITE

Images As Hyperlink : Just as text can act as a Hyperlink, so also images can act as hyperlinks.

External Link: links with other html document.

Internal Link: link in a same document. The anchor element is used to create both external and internal links.

Name Attribute: Named Anchor are used to create links to a specified section within document.

This is a two step process: At Top of page after Body Tag

At bottom of page before Body Tag **<A**

HREF="#LOCATION">Content

VIDYAPITH ACADEMY
MARQUEE

Marquee Tag: For scrolling text and image.

<Marquee>-----</Marquee>

Marquee Attributes

<Marquee behavior="slide"> </marquee>In this text scrolls for one

time only.

<Marquee behavior="alternate"> </marquee>In this text will scroll

continuously left to right.

<Marquee behavior="scroll"> </marquee>In this text will scroll continuously.

For direction of scrolling

<Marquee behavior="slide" direction="right"/"up"/"down">

For background color/width/height

<Marquee behavior="slide" bgcolor="red" direction="right" width="500" height=20>

For speed controlling in marquee

<Marquee scroll amount=1 maximum slow & 1000 maximum fast> <Marquee scroll delay=1000 maximum slow & 1 maximum fast>

MULTIMEDIA

Multimedia refers to "Multiple Medium" that is the ability to add sound and moving pictures to your webpages.

Adding Video : <embed src="movies.mpg" height="200" width="200" >

Adding Sound : Webpage contain basically two varieties of sounds that play whenever the visitor accesses the webpages that is the background sound, and when the visitor clicks something. The most common sound file formats are MIDI (Musical Instrument Digital Interface), WAV (as in Wave), AU (Audio) and (Audio Interchange File Format).

Sound files can be added in many of the ways. we can use the <A> anchor tag so that the visitor has the option whether to hear the sound or not. For example :

** Click Here For Background Music **

As the user clicks on the text, the background sound starts. So the visitor has the option whether to hear the sound or not.

Second option requires <Bgsound> tag : **<Bgsound src="Sound url">**

Here the src attribute specifies how many times the sound should play. If loop is specify loop = No or infinite to play the sound is an endless loop. **Adding**

Animated Gif's : (Graphics Interchange Format)

``

FORM Creation in HTML

Creating A Form : Webpages can be interactive if they collect information from the user. An HTML form is used for creating interactive web pages. The information entered by the user is passed back to a web server where it might be recorded in a Database for further use. So, a form is simply a collection of fields of information. These fields of information can be in the form of text boxes, radio buttons, pull down menus etc. Common uses of forms are Online order forms, Feedback forms, e-mail registration forms, or any web page in which input is required from the user in order to complete a given task. An HTML form consists of two parts. The first part, the user can see and which he fills out and next one is the part, the user can't see. The second part specifies how the server should process the user's information. Here we commonly use PHP script.

The Form Tag : A form must start with a `<FORM>` tag and end with a `</FORM>` tag. It has mainly two attributes:

1. Action
2. Method

Syntax: `<Form action="URL" method=get/post>Form`

Elements..... `</Form>`

Action is used to specify e-mail address that will receive the form information. If you have written a form handler

to which the data in the form will be sent, then you will have to add an `action="←RL"` attribute to specify the location of the handler.

The **Method** attribute tells the browser how to send the user's data to the server. There are only two possible values for the method attribute: `Method="get"` or `Method="post"`. The default value is `"post"`.

Using Form Control Elements : The form element can contain different elements to

create form controls.



Input Element : The <input>tag allows user to input information.

Page 63

Text: Used to create the single line text boxes.

Password: Similar to text,except the characters you type are not seen on screen.

Checkbox: As on or off switch.

Radio: Similar to a checkbox,except that in a group of radio buttons only one radio button can be on at a time.

Submit: Submits a form to be processed.

Reset: Clears all the input on a form.

Button: creates a button similar in appearance to a submit button.

File: Can be used to send a file from the surfer's computer along with the rest of the form data.

Form Creation:

Fieldset and legend tag : <Fieldset> tag is used to group more form controls together.

The <legend> tag, which is optional these tags require closing tag.The Syntax is :

<Fieldset>

<Legend>Group Label Text</Legend>

Control 1 Control 2 Control 3 Control 4

</Fieldset>

<html>

<head><title>Feedback Online Form</title></head>

<body bgcolor="black" text="white">

<form>

<fieldset>

<legend align="right">Form info@softbrain.org</legend>

<label>Full Address</label>



unique value for the name attribute for all radio buttons in a particular set.Syntax:



from the value attribute. The button size will be controlled directly by the text length.



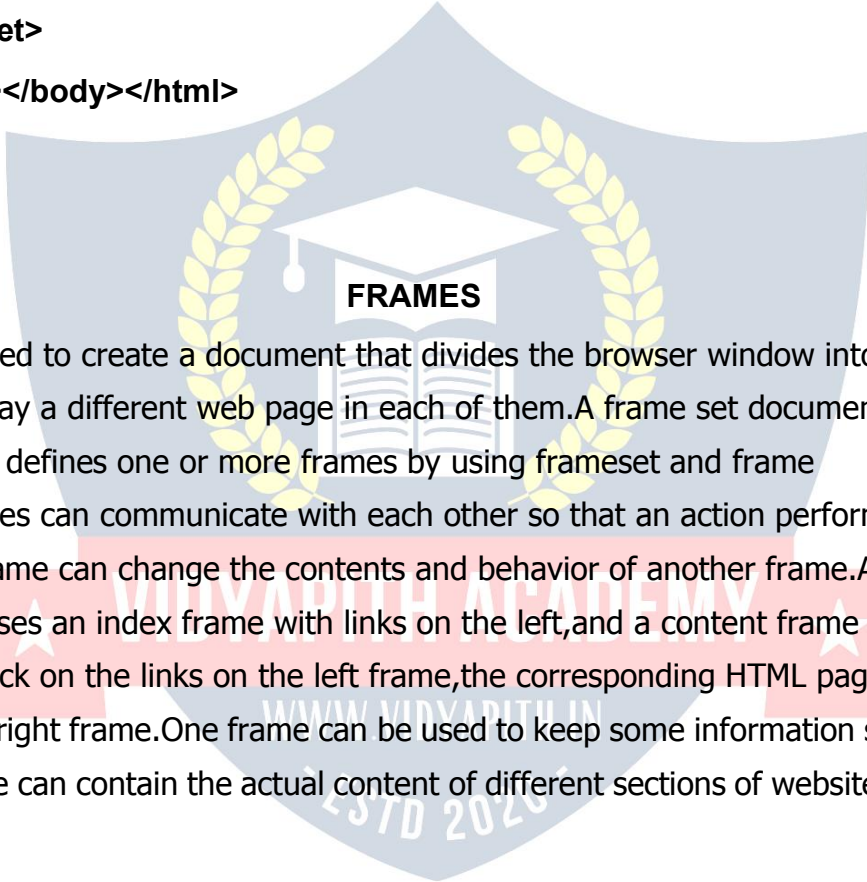
```
<input type="submit" name="sub" value="Submit">
```

Reset Button: The reset value of the type attribute cleans all form entries to the default once or leaves them blank if no defaults are specified. As the reset button can simply wipe off all the work done, therefore it should be well labeled and should never be placed before submit button.

```
<input type="reset" name="res" value="Reset">
```

```
</fieldset>
```

```
</form></body></html>
```



Frames are used to create a document that divides the browser window into different parts and display a different web page in each of them. A frame set document is a document that defines one or more frames by using frameset and frame elements. Frames can communicate with each other so that an action performed on a page in one frame can change the contents and behavior of another frame. A common frame layout uses an index frame with links on the left, and a content frame on the right. As you click on the links on the left frame, the corresponding HTML page will appear on the right frame. One frame can be used to keep some information static while the other frame can contain the actual content of different sections of website.

The <Frameset>Tag: The splitting of a browser screen into frames is accomplished with the <frameset> and </frameset> tags embedded into the HTML document. The <frameset> tag required one of the following two different attributes depending on whether the screen has to be divided into rows or columns.

Rows & Cols: Row attribute is used to divide the screen into multiple rows. Cols attribute is used to divide the screen into multiple columns. It can be set equal to a list of

values. Depending on the required size of each Rows and Columns. The value can be:



1. A number of pixels.
2. Expressed as a percentage of the screen resolution.
3. The symbol, '*' which indicates the remaining space.

Example: `<Frameset Rows="50%,50%"></Frameset>`

Result: It divides the browser screen into two equal horizontal sections.

`<Frameset Cols="50%,50%"></Frameset>`

Result: It divides the browser screen into two equal vertical sections.

The <frame>tag: Once the browser screen is divided into rows(horizontally)and columns(vertically),each different HTML document.This is achieved by using the <Frame>tag which takes in the following attributes:

Name="name": Gives the frame a unique name so it can be targeted by other documents.The name given must begin with an Alphanumeric character. **<Frame Name="AA">**

Src "Url": Indicates the URL of the document to be loaded into the frame.

Creating Framed Documents: Creating frames is actually quite simple.First of all you need to decide as how do you want to divide the browser's window of the webpage.Then write the html document that you want to put in the page segments.In addition to this a master document is to be written which is used to specify what goes where.Suppose we want to divide a page into two rows and four columns.Then the HTML code for the frameset will be:

`<html>`

`<head><title>FRAME ONLINE.... </title></head>`

`<frameset>`

`<frameset cols="30%,*">`

`<frameset rows="50%,50%">`


```
<frame name="aa">
```

```
<frame name="bb">
```

```
</frameset>
```

```
<frameset rows="50%,50%">
```

```
<frame name="cc">
```

```
<frame name="dd">
```

```
</frameset>
```

```
</frameset>
```

```
</frameset>
```

```
</html>
```

The **<Frameset>** and **<Frame>** tag: The master page doesn't have a **<body>** tag. Instead, it uses

<Frameset> tags. The **<Frameset>** tag is included after the **<Head>** tag and it establishes frames

within the HTML document. It does all the dividing. It requires a closing tag and there may be any number of frameset tags within an HTML document. The **cols** attribute is used for vertical frames whereas the **rows** attribute is used for horizontal frames. The width or height are specified as a comma-delimited list of sizes in pixels, percentage or as a proportion of the remaining space by using **"*"**.

Another basic tag is the **<FRAME>** tag. It contains the frame content and it tells the browser what to put in each frame. Each frameset must include a

<FRAME> definition for each division. As in our example there were four partitions and so four **<FRAME>** tags. Notice that frame does not have a closing tag.

Frame Border Attributes: By default, most browsers use a frame border. Border attributes that control the frame border. To turn off frame border set border=0 and for turn on border set border=8. You can use these attributes both in the <frameset> or <frame> tags. To remove all the borders from your frames add them in the <frameset> tag and to remove the border from individual frames include these attributes in the corresponding <frame> tag. You can also set the color of the border by just add bordercolor="purple".

Controlling Scroll Bars: Scrolling controls the appearance of horizontal and vertical scroll bars in a frame. This takes the value **Yes / No / Auto**.

In simple words, if the frame's content goes beyond the frame's width, most browsers add scroll bars so that the user can see the full content. Even though scroll bars are useful to see all the frame's content, sometimes they can be unnecessary and visually disturbing. You can control the use of scroll bars in a frame by including scrolling attributes.

Yes means the frame gets scroll bars—whether they are needed or not.

No means there will be no scroll bars, even if your frame content is much large. The browser will simply display as much as it can.

Auto is default If scroll bars are needed, they appear, if they are not needed they stay conveniently out of the way.

Targeting Named Frames: Whenever a hyperlink which loads a document in a frame is created, the file referenced in the hyperlink will be opened and will replace the current document that is in the frame.

In a situation where the new document needs to be opened in a different frame while keeping the document from which the new document was browser open in a different frame, a simple HTML coding technique must be used.

Since the hyperlink must open an HTML file in another frame, the frame in which the HTML file is to be opened need to be named. This is done by using the name attribute of the <FRAME> tags. The name takes one parameter, which is the frame name. The hyperlink tag will have to be supplied with the following information:

1. The filename.html file that has to be opened.

2. The name of the frame where the filename.html file has to be opened.

The attribute via which the frame name is specified is the TARGET attribute, which is a part of the <A> tag. This information is given as:

TARGET="framename"

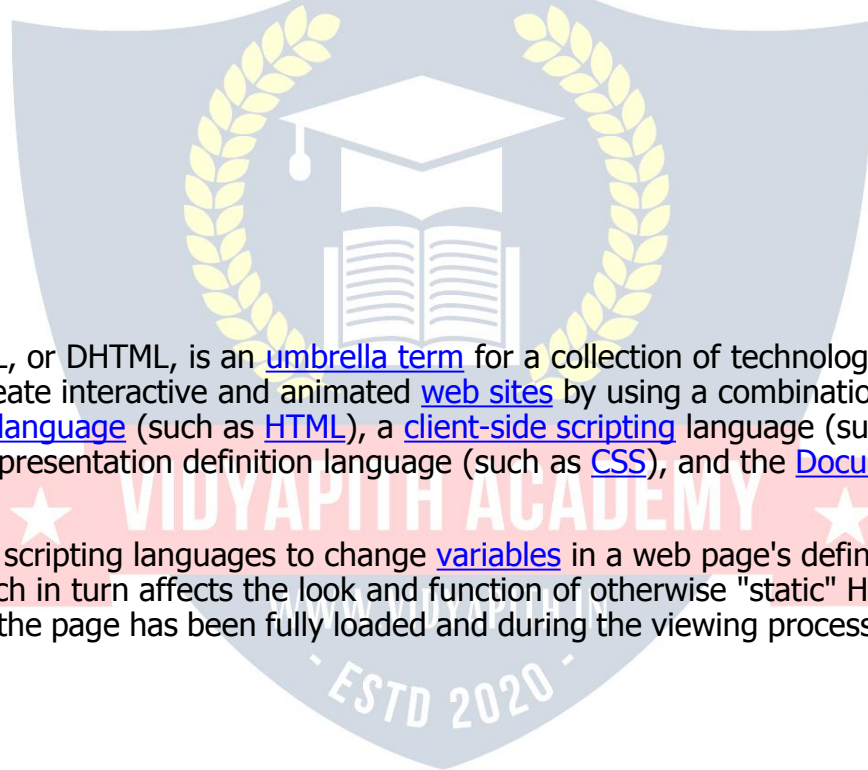
Link Content

DHTML

DHTML

Dynamic HTML, or DHTML, is an [umbrella term](#) for a collection of technologies used together to create interactive and animated [web sites](#) by using a combination of a static [markup language](#) (such as [HTML](#)), a [client-side scripting](#) language (such as [JavaScript](#)), a presentation definition language (such as [CSS](#)), and the [Document Object Model](#).

DHTML allows scripting languages to change [variables](#) in a web page's definition language, which in turn affects the look and function of otherwise "static" HTML page content, after the page has been fully loaded and during the viewing process. Thus the



dynamic characteristic of DHTML is the way it functions while a page is viewed, not in its ability to generate a unique page with each page load.

By contrast, a [dynamic web page](#) is a broader concept, covering any web page generated differently for each user, load occurrence, or specific variable values. This includes pages created by client-side scripting, and ones created by [server-side scripting](#) (such as [PHP](#), [Perl](#), [JSP](#) or [ASP.NET](#)) where the web server generates content before sending it to the client.

DHTML is differentiated from [AJAX](#) by the fact that a DHTML page is still request/reload-based. With DHTML, there may not be any interaction between the client and server after the page is loaded; all processing happens in Javascript on the client side.

By contrast, an AJAX page uses features of DHTML to initiate a request (or 'subrequest') to the server to perform actions such as loading more content.

USES

DHTML allows authors to add effects to their pages that are otherwise difficult to achieve. In short words: scripting language is changing the [DOM](#) and style. For example, DHTML allows the page author to:

- Animate text and images in their document, independently moving each element from any starting point to any ending point, following a predetermined path or one chosen by the user.
- Embed a ticker that automatically refreshes its content with the latest news, stock quotes, or other data.
- Use a form to capture user input, and then process, verify and respond to that data without having to send data back to the server. ➤ Include rollover buttons or drop-down menus.

A less common use is to create browser-based action games. Although a number of games were created using DHTML during the late 1990s and early 2000s, differences between browsers made this difficult: many techniques had to be implemented in code to enable the games to work on multiple platforms. Recently browsers have been converging towards the [web standards](#), which has made the design of DHTML games more viable. Those games can be played on all major browsers and they can also be ported to [Plasma](#) for [KDE](#), Widgets for [Mac OS X](#) and Gadgets for [Windows Vista](#), which are based on DHTML code.

The term "DHTML" has fallen out of use in recent years as it was associated with practices and conventions that tended to not work well between various web browsers. DHTML may now be referred to as [unobtrusive JavaScript](#) coding ([DOM Scripting](#)), in an effort to place an emphasis on agreed-upon best practices while [allowing similar effects in an accessible, standards-compliant way](#).

DHTML support with extensive DOM access was introduced with [Internet Explorer 4.0](#). although there was a basic dynamic system with [Netscape Navigator 4.0](#), not all HTML elements were represented in the DOM. When DHTML-style techniques became widespread, varying degrees of support among web browsers for the technologies

involved made them difficult to develop and [debug](#). Development became easier when [Internet Explorer 5.0+](#), [Mozilla Firefox 2.0+](#), and [Opera 7.0+](#) adopted a shared [DOM](#) inherited from [ECMAScript](#).

More recently, [JavaScript libraries](#) such as [jQuery](#) have abstracted away much of the day-to-day difficulties in cross-browser DOM manipulation.

Structure of a Web Page

Typically a web page using DHTML is set up in the following way:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>DHTML example</title>
  </head>
  <body>
    <div id="navigation"></div>

    <script>
      var init = function () {
        myObj = document.getElementById("navigation");
        // ... manipulate myObj
      };
      window.onload = init;
    </script>

    <!--
      Often the code is stored in an external file; this is done      by linking the file
      that contains the JavaScript.      This is helpful when several pages use the same
      script:
      -->
    <script src="myjavascript.js"></script>
  </body>
</html>
```

Document Object Model

DHTML is not a technology in and of itself; rather, it is the product of three related and complementary technologies: HTML, Cascading Style Sheets (CSS), and [JavaScript](#). To allow scripts and components to access features of HTML and CSS, the contents of

the document are represented as objects in a programming model known as the Document Object Model (DOM).

The DOM API is the foundation of DHTML, providing a structured interface that allows access and manipulation of virtually anything in the document. The HTML elements in the document are available as a hierarchical [tree](#) of individual objects, meaning you can examine and modify an element and its attributes by reading and setting properties and by calling methods. The text between elements is also available through DOM properties and methods.

The DOM also provides access to user actions such as pressing a key and clicking the mouse. You can intercept and process these and other events by creating event handler functions and routines. The event handler receives control each time a given event occurs and can carry out any appropriate action, including using the DOM to change the document.

Data Binding

Data binding is a DHTML feature that lets you easily bind individual elements in your document to data from another source, such as a database or comma-delimited text file. When the document is loaded, the data is automatically retrieved from the source and formatted and displayed within the element.

One practical way to use data binding is to automatically and dynamically generate tables in your document. You can do this by binding a table element to a data source. When the document is viewed, a new row is created in the table for each record retrieved from the source, and the cells of each row are filled with text and data from the fields of the record. Because this generation is dynamic, the user can view the page while new rows are created in the table. Additionally, once all the table data is present, you can manipulate (sort or filter) the data without requiring the server to send additional data. The table is regenerated, using the previously retrieved data to fill the new rows and cells of the table.

Another practical use of data binding is to bind one or more elements in the document to specific fields of a given record. When the page is viewed, the elements are filled with text and data from the fields in that record, sometimes called the "current" record. An example is a form letter in which the name, e-mail address, and other details about an individual are filled from a database. To adapt the letter for a given individual, you specify which record should be the current record. No other changes to the letter are needed.

Web Framework

Dreamweaver

Dreamweaver is a web authoring program that uses a “what you see is what you get” interface. The HTML code is being created and is accessible to you if you want to play with it.

Finally, let’s get started with Dreamweaver.

Launch Dreamweaver

1. Start Dreamweaver. You should see a startup screen.

Dreamweaver lets you publish your Web site by easily uploading your Web site to a server. Right now we won’t be worrying about that. For now, you will be creating and viewing your site locally (which means the site will be located on your computer).

We need to define your site:

Create a New Site

2. From the Site menu, choose Manage Sites... The Manage Sites Box should appear.
3. Click New and then choose Site from the pop-up menu.
4. Click on the Advanced Tab under Site Definition for Unnamed Site 1.

Local Info

Local Info is where we put in the information about the files that are on your local computer (which is where all your files are or will be for now).

5. In the Advanced Tab, under Category, choose Local Info.

6. Next to Site Name, type: My First Dreamweaver Web site. You will use this name when you go into Dreamweaver to retrieve the files associated with this site. This name has to be unique within Dreamweaver.
7. Next to Local root folder, click on the yellow folder icon to browse for where you want to store your web site on your local computer. You may wish to create a new folder to store your files in. I stored my Web site on F: (my flash drive) in a folder I called My WebSite. This is now your Root Folder
8. Click DONE. You should see a Files panel on the right side of your screen. The Local View in the Files Panel will display all the files for your Web site that are in the Root Folder on your computer. Later we will add information about connecting to the server. The Remote View shows which files have been uploaded to the server.
9. Once we've entered all this information about local and remote servers, it will be set up for us next time we use Dreamweaver on this computer. Unfortunately, it doesn't work that way on lab computers. If you move to another computer, you will have to set up your Web site on the new computer as well. 5

WHAT'S A ROOT FOLDER?

The local root folder is where you will keep ALL the files for your web site. You can create subfolders under it, but consider it to be the parent of all parent folders. You'll keep all of the files and subfolders for your Web site (including html pages, images, multimedia files, etc.) either in the root folder or a subfolder under the root folder..

Making your first Web Page:

10. Under the File menu at the top, choose New.
11. In the New Document dialog box, make sure you have Blank Page selected, the Page Type is HTML, and the Layout is <none>.
12. Click Create. You should now see a blank Web page in the Document Window.

13. Under the File Menu, choose Save.
14. In the Save As box, find your Web site folder and save your file as index.html
15. Click Save. In the Files panel, you should now see index.html. Congratulations!
You've created your first web page using Dreamweaver. Unfortunately, it's pretty boring at the moment.

Save the Page

RULES OF THUMB for naming web files:

- Do not use spaces or special characters, such as those found about the number keys on the keyboard, in your file names. You can use dashes and underscores. (e.g., my_page.html)
- Always use a 3- or 4-character file extension for every file you use on the Web. In the case of a regular Web page, always use .html or .htm.
- Be very specific about capitalizing because most web servers are case-sensitive.

Give your Web Page a Title In the document toolbar, find the Title box.

16. On the document toolbar, locate the Title field.
17. Type in a title for your page. Feel free to be creative (but not indecent). Hit Enter.
18. The title doesn't show up on the page, but will appear in tabs and across the top of browsers. Adding Some Content Add Content to the Page
19. In the Document window, type in a good title. This one will be the head of your Web page. If you can't think of anything, you can always use, "Aardvark Adoption Society of Delaware".
20. Add a subtitle. Again, it can be anything you want. If you'd prefer, it could be, "I speak for

Aardvarks who can't speak for themselves."

21. Now that you've got a good title, add some content. Logically, it should go with the title. Type one or two lines now for now. →e'll add more later. (If you can't think of anything, feel free to throw in some of the Aardvark content).

Now let's format the content:

22. On the Web page, highlight the title.

23. In the Property Inspector, click on the Format drop down arrow, and choose Heading 1.

Note: If you do not see the Property Inspector, choose Window >> Properties.

24. On the web page, highlight the second line of text.

25. From the Format menu on the Property Inspector, choose Heading 2.

Note: When formatting your text, keep in mind that text sizes in HTML are relative sizes. This means that,

when your page is viewed in a browser, the text size will depend on the default text size settings on each user's computer. Relative sizes give users the ability to adjust the text size to suit their individual needs.

This option is especially important for people with visual impairments, but it means that the appearance of your pages may vary from computer to computer. Playing with Font Properties

26. Highlight the Subtitle. Change the font size to 24. 7

27. Highlight any word on the page and click the Bold icon () on the Property Inspector. The selected word appears bold.

28. Highlight the title click the gray square to the right of the Size popup menu in the Property Inspector. A grid of colored squares appears, and the cursor changes to an eye dropper.
29. Click the eyedropper on the desired color. The field to the right of the color swatch reflects the code →eb browsers need to correctly display the color you've chosen.
30. Click anywhere on the page to deselect the text. The previously selected text appears in the new color.

Alignment

31. 1. Highlight the title and subtitle on the page.
32. Click the Align Center button () on the Property Inspector. The titles should now be centered on your page.

Creating Lists

33. First we need to add a list. 8
34. Type in a title for your list and then type in a list of items. Make sure there are at least 3 items in your list. Make sure you hit return after each item. (Feel free to use a list from the aardvark homework if you can't think of your own.)
35. Highlight the list title and choose Heading 3 from the Format menu on the Property Inspector.
36. Highlight the list items and click the Unordered List button on the Property Inspector.
37. The items appear as a bulleted list.
38. Now type in another List title and another list of items.
39. Highlight the title and select Heading 3.

40. Highlight the rest of the items and click the Ordered List button on the Property inspector (next to the ordered list button). The items should now appear in a numbered list.

41. Save your work. Get into the habit of saving your work regularly.

Page Properties

In this section you will learn how to set your page's background color, as well as the default text and link colors. These settings differ from what you've done so far on the Property Inspector because they'll apply to all the text and links on the page—not just a selected piece of text.

42. From the Modify menu at the top of the workspace, choose Page Properties.

43. To change the text color, click the bottom right corner of the text color box.

44. Using the eyedropper, choose a color by clicking on any of the color swatches.

45. Similarly, you can change the background color for the page using the Background color setting. 9

46. You can also change the color of links, active links, and visited links. An active link is a link just when the user clicks it. A color change lets the user know that s/he clicked on a link. A visited link is one that the user has already clicked on and visited. A color change lets the user know /he's already visited that link.

47. Select "Links" from the Category menu at left, then make your selections for link, visited link, and active link colors as you did in Step 3.

48. Click the Apply button to see how your color choices look on the index page. The Page Properties dialog box will remain on the screen.

49. →hen you're satisfied with the colors you've picked, click OK.

Note: You can add a background image if you want. Browse to find an image by the background image.

See what it looks like. Now you know that's an option. →e'll talk about this option more later. For now, go back to a solid color background.

Note: If you changed the color for any text using the Property Inspector, those changes will override general page color choices.

Web Page Preview

As you create a page in Dreamweaver, the page may not look exactly like it would in a browser. To preview the page in a browser:

50. Choose File >> Preview in Browser >> ie explore or Netscape or press the F12 key.

A web browser, such as Netscape or Internet Explorer, opens and loads the web page.

51. Take a look at your page to see how it will appear on the Web.

52. When finished, close the browser and switch back to Dreamweaver.

NOTE: Colored or underlined text can be mistaken for links. Choose your color scheme so that the links stand out from rest of the text. The text and links also should stand out from the background for readability. When selecting colors, keep it simple. Too many colors can be distracting or confusing.

Adding Images

Note: Any editing or resizing of an image should be completed before it is added to a web page.

53. First you need an image. Feel free to find one on your own that suits the theme of your web page. If you don't feel like it, I've included an image on my →eb site that you can download.

54. Next, add it to your Web page.

55. Place your cursor below the sub-header in your document and press Enter twice.
56. 2. Choose Insert >> Image. The Select Image Source dialog box will appear.
57. 3. Using the dialog box, locate and select an image, then click OK. 10
58. 4. You will first encounter a prompt to add accessibility attributes:
59. Type a brief description of the image in the Alternate text box, then click OK.
60. Because you're trying to use a file that lies outside your root folder (my website), you'll receive the following message:
61. Click Yes. The Copy File As dialog box will appear.
62. Click the Save button to place a copy of the image file inside your my website folder.
63. The image will appear on your web page.
64. Click the Refresh button on the Files panel () and the image file will appear in the list of local files.

Delete an Image

65. In the Document window, click on the image to select the image.
66. Press the Delete key. The image disappears.
67. To reverse the deletion, use the Undo command. From the Edit menu, choose Undo. The image should re-appear on the page.

Create a Second Page

Next, you will create a second page for your Web site.

68. From the File menu, choose New...The New page dialog box appears.
69. Click the Create button. The new file will appear in the document window.
70. Save your new page in the my website folder as Page2.html

71. In the Title field, type an appropriate title for this page. →e'll eventually make it into a Calendar of Events Page, if that helps you decide on a title.

72. Save your work.

Page Planning: When planning your Web site, you will need to decide how to organize each Web page. What will be placed on each page. Divide your site into pages for simple access. If your readers want to read or print only a calendar of events, you don't want them to have to print the entire site to see it. Your site planning will also include how the pages will be linked together.

Page Navigation

Now that we have two pages, you can link them together. →e'll link the index.html page to the Page2.html and vice versa. It's easy in Dreamweaver!

73. Use the tabs on the Document toolbar to toggle back to the index page.

74. On the index page, type some descriptive text that describes your Page2.html.

75. Highlight the text you want to make a link.

76. In the Property Inspector, click on the Browse for File icon to the right of the link box. A Select File dialog box appears. 12

77. Using the dialog box, locate and select the file Page2.html.

78. Click the OK button. Cool! You just made a link!

79. The selected linked text is underlined and appears in the link color you chose. In the Property Inspector, the Link field reads: "Page2.html

80. Save your work and preview the page in a browser.

81. Test the link. When finished, return to Dreamweaver.

82. Now, try linking Page2.html back to the index page.

83. Save your work and test the new link in a browser.

Note: The text you use for your link should be brief but descriptive, giving your visitors an idea of where the link will take them even before they click it.

Let's compare: [click here for more information lacks detail.](#)

Compared to [Calendar of Upcoming Events](#) , which is clear and concise.

84. Link to a Web Page

85. Next let's add an external link. By that I mean let's link to a page that is located somewhere on the world wide web. We'll need to use the full (absolute) URL to link to this page.

86. Go to the bottom of your index.html page and type "More info on Aardvarks".

87. Highlight the text you just typed.

88. In the Link box in the Property Inspector, type <http://en.wikipedia.org/wiki/Aardvark/> and press Enter.

89. The text you selected should now link to Wikipedia's web page on Aardvarks. To make sure the link works properly, preview the page in a browser.

90. Switch back to Dreamweaver to continue editing your page.

Add an E-mail Link

Now you'll make your e-mail address a link so that visitors can send you questions and comments via email.

91. At the bottom of your page, type "Send me an email" and highlight the word email.

92. From the Insert toolbar (across the top, below File Edit, etc.), click on the envelope icon to insert an email link.

93. Place your cursor in the E-mail box and type your email address:
94. Hit OK. Your e-mail address appears as a link.
95. Save your work. Please note that the email link will try to open a desktop email program if you have one on your computer. However, if the computer you're working on doesn't have an email program set up, testing this link won't work.

Note: One of the advantages in creating a course Web site is adding appropriate links to other sites on the Web. You may want to compile a list of links that support your instructional goals. This allows you to supplement your page with appropriate Web resources. Plus it's always fun to look around at other relevant pages to learn new things about your topic and create your own relevant Weblibliography.

Creating a Link within a Page

the Target of the link

96. Place your cursor at the top of your page
97. Choose Insert >> Named Anchor. The Named Anchor dialog box will appear.
98. In the Anchor Name box, type a name for the anchor. This should describe the content in that section of the page.
99. Click OK. A yellow anchor will appear just before the text. If the yellow anchor symbol does not appear, then choose View >> Visual Aids >> Invisible Elements.

Create a link to the anchor

100. At the bottom of the page, type "Back to Top". Highlight this text.
101. On the Property Inspector, click on the point-to-file icon to the right of the Link field and drag it to the schedule anchor symbol. (You literally click on it and, without releasing the mouse, drag it over the anchor symbol in your Web page).

As the arrowhead moves over the anchor, the anchor name preceded by the pound sign (#) will appear in the link field. The selected text is now a link, and is underlined with a different text color.

The form that this type of link uses is different from a standard link. In the link field, a pound sign appears followed by the name of the anchor. The # (pound sign) indicates that the link is to an anchor on the Web page.

Note: When planning your Web site, you will need to decide how to organize links between all the pages.

A good practice is to place links to other pages at the top and the bottom of each page. Also, each page should be linked to your home page (index.html).

Page Layout with Tables

We can use Tables to lay out the format of text and images. A table has rows and columns and may be used for layout in a number of ways, including placing text or images side by side on a page. In this section, you'll make a table that shows the Calendar dates and events for an Aardvark Sanctuary.

Create a Table

102. Place your cursor at the top of Page2.html after the text. Now, you'll insert a table with 3 rows and 2 columns.

103. From the Insert menu, choose Table. The Table dialog box appears.

104. Under Table size, adjust the settings as follows:

Rows = 3

Columns = 2

Table width = 75 %

Border thickness = 1 15

105. Click OK. The table will appear on your page.

106. Place your cursor in the upper-left cell of the table and type: "Schedule of Events"

107. On the second row, in the left-hand cell type: Date

108. Press the [Tab] key to move to the next cell in the table.

109. Type: Event

110. Press the [Tab] key again to move to the next row.

111. Type in a date. Move to the next cell, and type in a description of an event. As you type in a long paragraph, the words wrap within a cell.

112. Press the TAB key, and a new row is automatically added to the table. Enter the date and event description in the fourth row.

Merge Cells

We want the Schedule of Events to be centered across the entire table. To do that, we first need to merge the two top cells in the table.

113. Click and drag your cursor from left to right across row 1. ↔ou'll now merge those two cells into one.

114. Choose Modify >> Table >> Merge Cells. Now, in row 1, you should see one large cell.

Insert an Image

115. Place your cursor in the right-hand cell of the last row. Press the [Tab] key to add a new row.

116. Place your cursor in the right-hand cell of that new row. Choose Insert >> Image. The Select Image Source dialog box appears.

117. Navigate to an image – either one you’ve downloaded and is on your flash drive or one from my Web site that you can download.

118. Click OK. The image you chose will appear in the table cell.

Add a new row to the table.

After you have created a table, you may need to add information in the middle of the table. You can add rows or columns to expand the table.

119. Click in the third row of your table.

120. Choose Modify >> Table >> Insert Row.

A new row appears in the table.

121. Click in the second column of the new row and insert another image (your choice). The image appears in the cell.

122. Click in the cell to the left of the image and type some relevant information in the cell.

Delete a row

The table can also be adjusted by removing unnecessary rows or columns.

123. Click in the second row of your table.

124. From the Modify menu, choose Table, and then choose Delete Row. The row and its contents are deleted. Table

Resize columns and rows

Depending on the content you add to your table you might find that columns are too wide or narrow.

You can adjust the column width to better fit the content in your table.

125. Place the cursor over the line dividing the columns. The cursor turns into a twoheaded arrow.

126. Click and drag the two-headed arrow toward the right to the middle of the table. The columns are re-sized.

127. Place the cursor over the right side of the table. Click and drag the two-headed arrow to resize the table again.

128. Save and preview the page.

Try changing the size of the browser window. Notice that the table size changes as the browser window size changes – why? The reason the table size is not constant is because the width of the table is expressed as a percentage of the total width of the browser window, in this case, 75%. If you wish to lock the table to a specific size, you have to express the table dimensions differently.

Table dimensions can also be expressed as fixed using “pixels”. A pixel is a single point in a graphic image. Screens are divided into thousands (or millions) of pixels, arranged in rows and columns. On a 17" web page, a recommended pixel size for a table's width would be no higher than 600. (Assuming a screen resolution of 800 x 600).

When your table is selected, you can view and change the table width on the Property Inspector.

Experiment with cell color and other options

129. To add background shading to a cell or row, select the cell(s) and click on the bottom corner of the Bg color box on the Property Inspector

130. Click on a swatch with the eyedropper to choose a color.

131. Try various formatting options for the rest of the table using the Property Inspector. Change the size, color or style of the text, and try the alignment options.

132. Save your work and preview.

The logo for Vidyapith Academy is a circular emblem with a yellow laurel wreath. In the center is an open book with a graduation cap above it. Below the emblem is a red banner with white stars at each end. The text 'VIDYAPITH ACADEMY' is written in white on the banner, and 'PROGRAMMING LANGUAGE C' is written in black below it. The website 'WWW.VIDYAPITH.IN' and 'ESTD 2020' are also visible.

PROGRAMMING LANGUAGE C

Why we use Language ?

We use language due to following reasons : ➤

It stores data

- How to interact between input and output
- The operator uses to transform and combine data.

What is C ?

By 1960 languages come into existence. COBOL was being used for commercial purposes, FORTRAN for Engineering and Scientific applications. An International committee was setup to develop languages. This committee came out with a language called ALGOL60. ALGOL-60 never really became popular because it seemed too

abstract, too general. To reduce the abstractness and generality a new language called Combined Programming Language (CPL) was developed at Cambridge University. CPL was an attempt to bring ALGOL 60 new version, but CPL turned out to be so big, having so many features, that it was hard to learn and difficult to implement. BCPL (Basic Combined Programming Language) developed by Martin Richards at Cambridge University aimed to solve this problem by bringing CPL down to its basic good features. But unfortunately it turned to be less powerful and too specific. Dennis Ritchie inherit the features of CPL and BCPL and made a language named C. C forms the basis for many advanced, highly powerful and effective programming languages. C is a programming language developed at AT&T's Bell Laboratories of ←SA in 1972. It was designed and written by a man named Dennis Ritchie. In the late seventies C began to replace the more familiar languages of that time like PL/I, ALGOL etc. No one pushes C. C seems so popular, because it is reliable, simple and easy to use. The concept of C derives its origin from a primitive form of C which was called Basic Combined Programming Language (BCPL) developed by Ken Thompson of Bell Laboratories which he referred to as 'B'. By now it might not be difficult to guess that a name as cryptic as C was conferred to it because it was considered to be a modified more adaptive successor of "B". C's compactness and coherence is mainly due to the fact that it's a one man language.

C Basics

Before we embark on a brief tour of C's basic syntax and structure we offer a brief history of C and consider the characteristics of the C language.

In the remainder of the Chapter we will look at the basic aspects of C programs such as C program structure, the declaration of variables, data types and operators. We will assume knowledge of a high level language, such as PASCAL.

It is our intention to provide a quick guide through similar C principles to most high level languages. Here the syntax may be slightly different but the concepts exactly the same.

C does have a few surprises:

- Many High level languages, like PASCAL, are highly disciplined and structured.
- However beware -- C is much more flexible and free-wheeling. This freedom gives C much more power that experienced users can employ. The above example below (mystery.c) illustrates how bad things could really get.

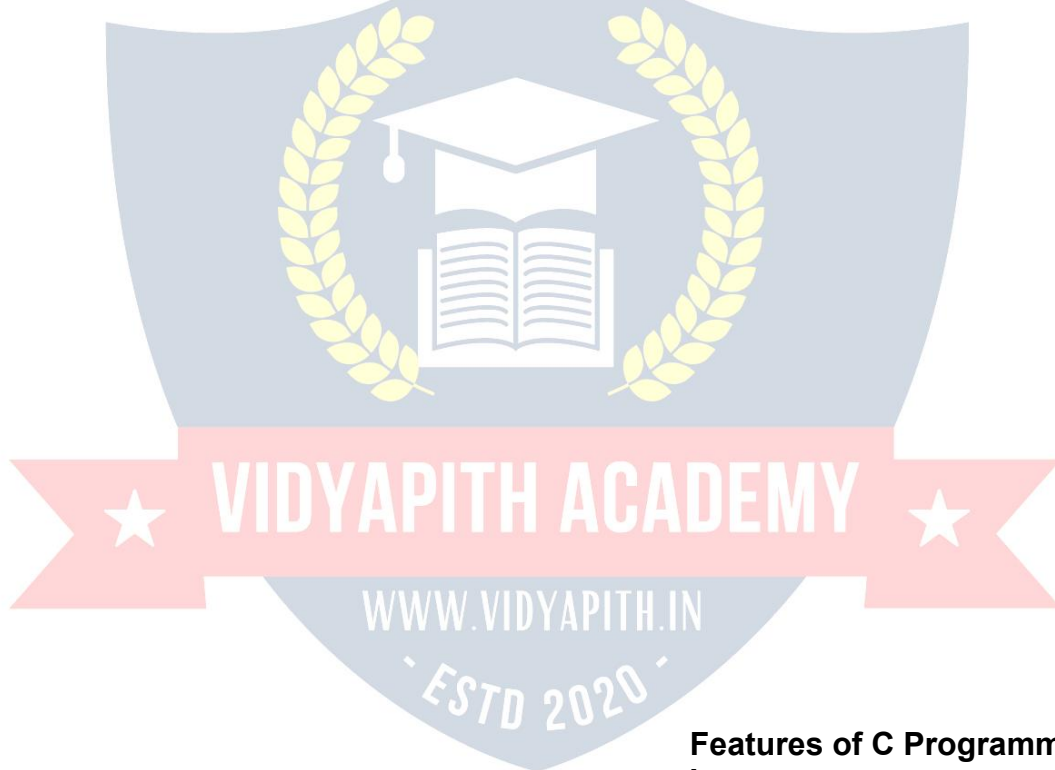
Characteristics of C

We briefly list some of C's characteristics that define the language and also have lead to its popularity as a programming language. Naturally we will be studying many of these aspects throughout the course.

- Small size
- Extensive use of function calls
- Loose typing -- unlike PASCAL
- Structured language
- Low level (BitWise) programming readily available
- Pointer implementation - extensive use of pointers for memory, array, structures and functions.

C has now become a widely used professional language for various reasons.

- It has high-level constructs.
- It can handle low-level activities.
- It produces efficient programs.
- It can be compiled on a variety of computers.



Features of C Programming Language :

C Programming is widely used in Computer Technology, We can say that C Programming is inspiration for development of other languages. We can use C Programming for different purposes. Below are some of the Features of C Programming language -

1 . Low Level Features :

Features of C	
Low Level Language Support	Program Portability
Powerful and Feature Rich	Bit Manipulation
High Level Features	Modular Programming
Efficient Use of Pointers	

1. C Programming provides low level features that are generally provided by the Lower level languages. C is Closely Related to Lower level Language such as "**Assembly Language**".
2. It is easier to write assembly language codes in C programming.

2 . Portability :

1. C Programs are portable i.e they can be run on any Compiler

with Little or no Modification

2. Compiler and Preprocessor make it Possible for C Program to run it on Different PC

3 . Powerful

1. Provides →ide verity of '**Data Types**'
2. Provides →ide verity of '**Functions**'
3. Provides useful Control & Loop Control Statements

4 . Bit Manipulation

1. C Programs can be manipulated using bits. We can perform different operations at bit level. We can manage memry representation at bit level.
2. It provides wide verity of bit manipulation Operators. We have bitwise operators to manage Data at bit level.

5 . High Level Features :

1. It is more User friendly as compare to Previous languages. Previous languages such as BCPL,Pascal and other programming languages never provide such great features to manage data.
2. Previous languages have there **pros and cons** but C Programming collected all useful features of previous languages thus C become **more effective language**.

6 . Modular Programming

1. **Modular programming** is a software design technique that increases the extent to which software is composed of separate parts, called **modules**



2. C Program Consist of Different Modules that are integrated together to form complete program

7 . Efficient Use of Pointers

1. Pointers has direct access to memory.
2. C Supports efficient use of pointer .

C Program Structure

A C program basically has the following form:

- Preprocessor Commands
- Type definitions
- Function prototypes -- declare function types and variables passed to function.
- Variables
- Functions

We must have a main() function.

A function has the form:

```
type function_name (parameters)  
{  
    local variables  
    C Statements  
}
```

If the type definition is omitted C assumes that function returns an **integer** type. **NOTE:** This can be a source of problems in a program.

So returning to our first C program:

```
/* Sample program */  
  
main()
```

```
{  
  
    printf( ``I Like C n" );  
    exit ( 0 );  
  
}
```

NOTE:

- C requires a semicolon at the end of **every** statement.
- printf is a **standard** C function -- called from main.
- n signifies newline. **Formatted output** -- more later.
- exit() is also a standard function that causes the program to terminate. Strictly speaking it is not needed here as it is the last line of main() and the program will terminate anyway.

Let us look at another printing statement:

```
printf( ``.n.1n..2n...3n");
```

 The

output of this would be:

```
.  
.1  
..2  
...3
```

C Keywords

Keywords are the set of predefined words whose functionality has been expressed to the compiler and whenever called upon they furnish their task with utmost comfort. The keywords cannot be used for any function other than what it is defined for, not even as variable names. This code of conduct it will result in giving a new meaning to the keyword, which is not permissible by the computer . There are 32 keywords in C which are used at various points in a program to launch the proficiency of the language.

C Instructions

The different types of constants, variables and keywords the next logical step is to learn how to learn they are combined to form instructions. There are four types of instruction in C :

- **Type Declaration Instruction**
- **Input/ Output Instruction**
- **Arithmetic Instruction**
- **Control Instruction**

Type Declaration Instruction : We can declare the type of variables used in C Program by using this instruction. Any variable used in a program must be declared before using it any statement. The type declaration statement usually written at the beginning of the C program.

eg. `int EmpCode;`
`float Bsalary;`
`char EmpName`

Input/ Output Instruction : We can perform the function of supplying input data to a program and obtaining the output results from it.

Arithmetic Instruction: We can perform arithmetic operations between constants and variables.

Control Instruction : We can control the sequence of execution of various statements in a C program.

Variables

A program is made of data and instructions to manipulate those data. Note that data have to be stored somewhere, and thus will need some memory space in the RAM.

A variable is an entity that is used to store data. Without variables, there is no way (or actually NO PLACE) to store data. A variable has

- a name (more specifically a symbolic name)
- an associated physical memory space (portion in a RAM) ➤ a data type
- a value (depends on data type)
- a scope
- a lifetime

How do you declare a variable?

In C, the syntax for declaring a variable is as follows:

<variable name>

The symbol <item> means that it is required to specify the item enclosed within a pair of angled brackets. A semicolon signifies the end of a declaration. A missing semicolon will cause the compiler to generate a syntax error. Variables should be named following the C naming conventions.

Example:

```
char ch;  
int i; float  
f;
```

```
double d;
```

It is possible to declare several variables of the same type on the same line. In such a case, a comma should be inserted between two variables. A missing comma will generate a syntax error.

Example:

```
char ch1, ch2; int
```

```
x, y, z;
```

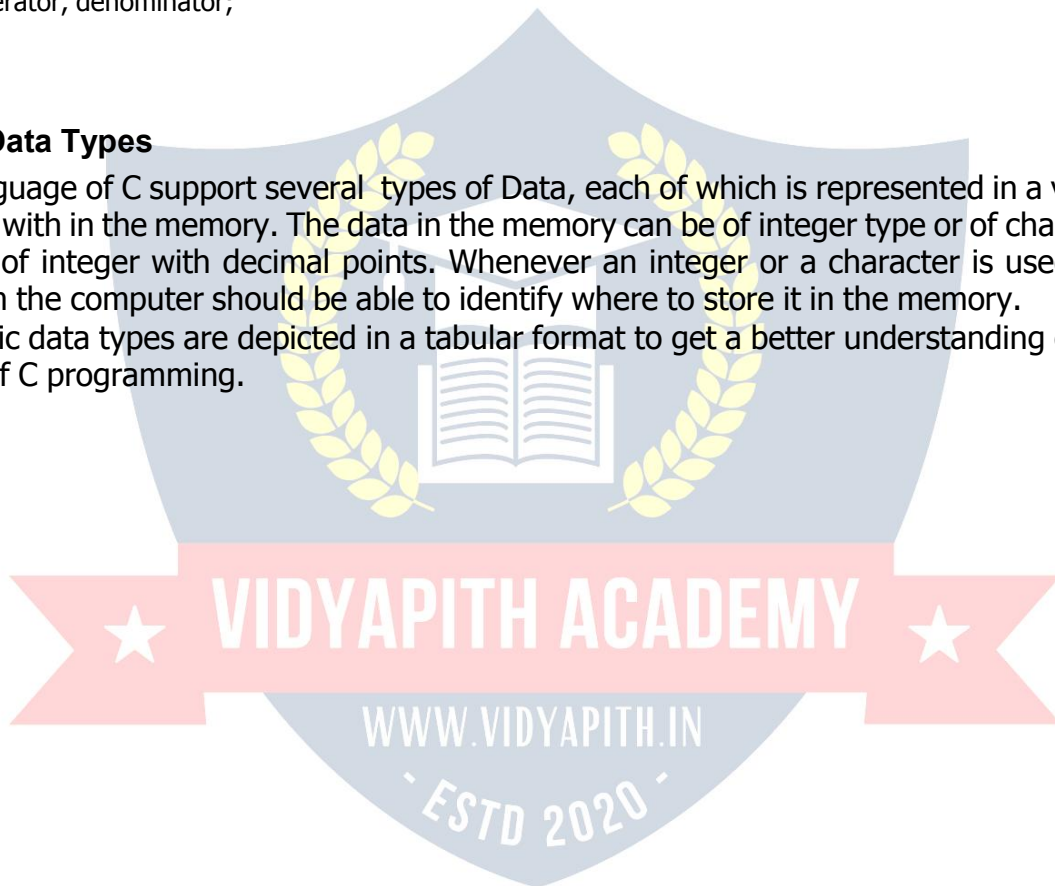
```
float hourly_rate, number_of_hours, salary; double
```

```
numerator, denominator;
```

Basic Data Types

The language of C support several types of Data, each of which is represented in a varied manner with in the memory. The data in the memory can be of integer type or of character type or of integer with decimal points. Whenever an integer or a character is used in a program the computer should be able to identify where to store it in the memory.

The basic data types are depicted in a tabular format to get a better understanding of the basics of C programming.



Data Type	Description	Memory Requirement	Range	Format Specifier
Int	whole numbers	2 bytes	-32768 to	%d, %i
long	-	4 bytes	+32767	%ld
Char	Characters	1 Byte	0 to 255	%c, %s
Float	Numbers with Decimals	4 Bytes	1.0E-37 to 1.0E+37	%f
Double	Numbers with	8 Bytes	1.7E-308 to	%lf

Int : The particular data type deals in storing only whole integer numbers and they are incapable of dealing with fractions or number with decimals. The number which they can deal with a range from -32768 to +32767. An int requires 2 bytes of memory by most computers.

Char : The char data type holds individual characters or alphabets and they require only 1 byte of memory. Most compilers permits a range of 0 to 255 regarding the char data type.

Float : A float data type is involved in restoring floating point numbers or to put it simply, it is used to store numbers with decimal point . The float desires 4 bytes for storage and has a fractional precision of 6 digits.

Double : A double data type is also used for the restoration of floating point numbers. The difference lies in the precision and storage space. Double offers a great precision, of about 15 digits and compiler requires 8 bytes to store a double. The ranges with in which the float can be represented are 1.7 E - 308 to 1.7 E + 308 resp.

Qualifiers Used With Basic Data Types

To expand the horizon of the basic data types, making it more flexible and adaptable to get into various situations with preciseness and accuracy. The qualifier tagged to the data types are **LONG, SHORT, SIGNED AND UNSIGNED**. To make any data type a qualifier, the name of the qualifier has to be tagged before the data type.

Long : As the name suggests the qualifier long extended the range of the data type to which it is prefixed. It can be applied to only the int and double data types and it expands the spectrum of range from a maximum of 2,147,483,647 to -2.147,483,647. A long int requires memory that is more than an ordinary int and therefore takes 4 bytes for storage. To make an integer a long qualifier just add the word long before it.

Short : This qualifier is most interesting because if an ordinary int has the same memory value as that of long i.e. 4 bytes a short will have 2 bytes, but if an int has memory of 2 bytes a short will have the same memory i.e. 2 Bytes.

Signed : By using signed data qualifier that data type can accommodate both positive and negative numbers. When a variable is signed it has the same positive and negatives ranges as a regular int. By default an int is a signed data type.

Unsigned : The qualifier unsigned is tagged mostly to int and it also has the same memory requirement as that of ordinary int i.e. 2 bytes. Unsigned integer have a range from 0 to 65535.

Constants

ANSI C allows you to declare **constants**. When you declare a constant it is a bit like a variable declaration except the value cannot be changed. The const keyword is to declare a constant, as shown below:

```
int const a = 1; const  
int a =2;
```

Note:

- You can declare the const before or after the type. Choose one and stick to it.
- It is usual to initialise a const with a value as it cannot get a value **any other way**.

The preprocessor #define is another more flexible (see Preprocessor Chapters) method to define **constants** in a program. You frequently see const declaration in function parameters. This says simply that the function is **not** going to change the value of the parameter. The following function definition used concepts we have not met (see chapters on functions, strings, pointers, and standard libraries) but for completeness of this section it is included here:

```
void strcpy(char *buffer, char const *string)
```

The second argument string is a C string that will not be altered by the string copying standard library function.

Operators in C :

Operators can briefly be defined as the tools used for solving various mathematical, conditional, relational and logical problems. The operators that you will be dealing with are arithmetic operators , unary operators, relational and logical operators . The item in between which the operators are placed are called operands.

Arithmetic operators :

Arithmetic Operators can be considered the main point of all the operators. Arithmetic operators are tools that help us in computing various mathematical operations.

Operators	Function Performed
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	(Modulus) Finds the remainder

Unary Operators :

There are two types of unary operators :-

- **The increment operator (denotes as ' ++ ')**
- **The decrement operator (denotes as ' -- ')**

When the increment operator is prefixed to a variable which holds an integer it increases the value of the number by one. Similarly the decrement operator decreases the value by one.

Relational and Logical Operators :

Operators	Significance
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
Equality Operators	
==	Equal to
!=	Not equal to
Logical Operators	
&&	And
	Or
!	Not

Assignment is = *i.e.* `i = 4; ch = 'y';`

Increment ++, Decrement -- which are more efficient than their long hand equivalents, for example:-- `x++` is faster than `x=x+1`.

The `++` and `--` operators can be either in post-fixed or pre-fixed. With pre-fixed the value is computed before the expression is evaluated whereas with post-fixed the value is computed after the expression is evaluated.

In the example below, `++z` is pre-fixed and the `w--` is post-fixed:

```
int x,y,w;
main()
{
x=((++z)-(w--)) % 100;
}
```

This would be equivalent to:

```
int    x,y,w;
main()
{
z++;
x=(z-w) % 100;
w--;
}
```

The `%` (modulus) operator only works with integers. Division `/` is for both integer and float division. So be careful. The answer to: `x = 3 / 2` is 1 even if `x` is declared a float!!

RULE: If both arguments of `/` are integer then do integer division.

So make sure you do this. The correct (for division) answer to the above is `x = 3.0 / 2` or `x= 3 / 2.0` or (better) `x = 3.0 / 2.0`. There is also a convenient shorthand way to express computations in C. It is very common to have expressions like: `i = i + 3` or `x = x*(y + 2)` This can written in C (generally) in a **shorthand** form like this:

expr₁ op = expr₂

which is equivalent to (but more efficient than):

$expr_1 = expr_1 \text{ op } expr_2$

So we can rewrite $i = i + 3$ as $i += 3$ and $x = x*(y + 2)$ as $x *= y + 2$.

NOTE: that $x *= y + 2$ means $x = x*(y + 2)$ and NOT $x = x*y + 2$.

Comparison Operators

To test for equality is `==`

A warning: Beware of using ```="` instead of ```=="`, such as writing accidentally

`if (i = j)`

This is a perfectly **LEGAL** C statement (syntactically speaking) which copies the value in "j" into "i", and delivers this value, which will then be interpreted as TRUE if j is nonzero. This is called **assignment by value** -- a key feature of C.

Not equals is: `!=`

Other operators `<` (less than), `>` (greater than), `<=` (less than or equals), `>=` (greater than or equals) are as usual.

Pointers

C supports the use of pointers, a type of reference that records the address or location of an object or function in memory. Pointers can be dereferenced to access data stored at the address pointed to, or to invoke a pointed-to function. Pointers can be manipulated using assignment or pointer arithmetic. The run-time representation of a pointer value is typically a raw memory address (perhaps augmented by an offset-within-word field), but since a pointer's type includes the type of the thing pointed to, expressions including pointers can be type-checked at compile time. Pointer arithmetic is automatically scaled by the size of the pointed-to data type. Pointers are used for many different purposes in C. Text strings are commonly manipulated using pointers into arrays of characters. Dynamic memory allocation is performed using pointers.

Statements in C Language

C needs some kind of managerial system which instructs the computer what function has to be performed at which time. The control statement act as a manager. You will amazed at the functions the different control statements handle. For example in a program you may have a statement in which a multiplication operation is to be performed and in the next an addition operation and after the next line there may be a statement which may want itself to be repeated 8 times and another statement may like itself to be skipped and attended to at a later stage. All these conditions are managed by the control statements.

The sequence control structure is one of the basic uncomplicated control statements. The idea is very simple. In the sequence control instructions the statements to be executed are done so in a serial manner one after the other. The C compiler first executes the opening line and then moves on to the next and so on. In a serial fashion and the entire program is executed in this manner.

Conditional Statement

C program executes program sequentially. Sometimes, a program requires checking of certain conditions in program execution. C provides various key condition statements to check condition and execute statements according conditional criteria. These statements are called as '**Decision Making Statements**' or '**Conditional Statements**'. Followings are the different conditional statements used in C :

1. **If Statement**
2. **If-Else Statement**
3. **Nested If-Else Statement**
4. **Switch Case**

If Statement: The if statement can be used to test conditions so that we can alter the flow of a program. Code:

```
#include<stdio.h>
int main()
{
    int mark; char pass;
    scanf("%d",&mark);
    if (mark > 40)
        pass = "y";
    return 0;
}
```

If-Else Statement:The if statement first tests if a condition is true and then executes an instruction and the else is for when the result of the condition is false. Code:

```
#include<stdio.h> int
main()
{
    int mark; char pass;
    scanf("%d",&mark);
    if (mark > 40)
    {
        pass = "y";
        printf("You passed");
    }
    else
    {
        pass = "n"; printf("You
        failed");
    }
    return 0;
}
```

The Switch Statement:The switch statement is just like an if statement but it has many conditions and the commands for those conditions in only 1 statement. It is runs faster than an if statement. In a switch statement you first choose the variable to be tested and then you give each of the conditions and the commands for the conditions. You can also put in a default if none of the conditions are equal to the value of the variable. **For Loop, While Loop, Break and Continue**

Control structures are basically of three types –

- **Sequence statements**
- **Iterative statements**
- **Selection statements**

Sequence Statements : All the State in a program except the iterative & statements. They are generally the individual statements which performs the task of input, output, assignment declaration etc.

Iterative Statement are those repeated execution of a particular set of instructions desired number of times. These statements are generally called loops for their execution nature.

Types of Looping Statements:

Basically, the types of looping statements depends on the condition checking mode. Condition checking can be made in two ways as : Before loop and after loop. So, there are 2(two) types of looping statements.

- **Entry controlled loop**
- **Exit controlled loop**

1. Entry controlled loop :

In such type of loop, the test condition is checked first before the loop is executed.

Some common examples of this looping statements are :

- **while loop**
- **for loop**

2. Exit controlled loop :

In such type of loop, the loop is executed first. Then condition is checked after block of statements are executed. The loop executed atleast one time compulsarily.

Some common example of this looping statement is :

- **do-while loop**

In C language the iterative statements (loops) can be implemented in the three loops and they are

The For Loop Syntax-

for (initialization; condition; incrementation)

```
{  
    .....body of loop.....  
}
```

For Loop will perform its execution until the condition remains satisfied. If the body of the loop consists of more than one statement then these statements are made compound by placing the open and closed curly brackets around the body of the loop. For loop is a count loop. The initialization condition and increamentation may be done in the same statement. For loop will not execute at least once also if the condition is false at the first time itself.

The While Loop

Syntax –

```

Initialization;
While (condition)
{
    Body      of      loop;
Incrementation;
}

```

In this loop, initialisation, condition and incrementation is done in the three different statements. This loops is count as well as event loop. In case of while loops the body of the loop will consist of more than one statements because each time one statement will be of incrementation. Hence the open and closed curly brackets are required.

Do-While Loop Statement:

The third loop statement available in C is do-while statement syntax :- Initialization;

```

Do
{
    Body of loop;
Incrementation;
}while(condition)

```

An introduction to Arrays

The concept of arrays depends on other data types, which was meant to facilitate the storage of abundant amount of number and wasting an entire delivery of space in the memory. It would be like buying a two dozen books and ultimately using one and dumping the rest in the garbage. Arrays can be defined as a collection of similar elements. An array of elements can be formed only if all the elements are of one particular data type i.e. all of them are either integer or character or a floating point number, but there cannot exist an array which hosts an combination of these data types.

Advantage of Arrays

The usage of arrays is crucial to any professional programming. Arrays just make the life of a programmer a bit easier. It relieves him of a burden of having to keep naming the variables. An array is a single variable holding multiple data. Each data is referred to by its element number, If for instance, you wish to store a list of Items you could just name the variable 'item_code' and keep storing the data by referring to item_code (1), item_code (2),item_code (3),..... etc.

Array Declaration

Like other variables an array needs to be declared so that the compiler will know what kind of an array and how large an array we want. In our program we have done this with the statement **int sal [20] ;**

Here int specifies the type of the variable, just as it does with ordinary variables and the word **sal** specifies the name of the variable. The [20] however is new. The number 20

tells how many elements of the type **int** will be in our array. This number is often called the "dimension" of the array. The bracket ([]) tells the compiler that we are dealing with an array.

Types of Arrays

The array are divided in to two parts :

- **Single Dimensional Arrays**
- **Multidimensional Arrays**

Single Dimensional Arrays :

The single dimensional arrays as the name suggested, handles only a single advance of similar elements. That is to say you can only have one row of elements and the size of a single dimensional array depends on the programmer. Basically the single dimensional arrays facilitate the accommodation of many similar elements in a single variable. The succeeding sections after which you will have a cloud less view of the single dimensional arrays.

Multidimensional Arrays :

The multidimensional arrays are two or more arrays clubbed together. They can be of two dimensional type or three dimensional types or more, but you can restrict the research only to two and three dimensional arrays. An example of the two dimensional arrays is the matrix because like the matrix the two dimensional arrays have two matrices one symbolizing the row elements and the other the column elements. The three dimensional array is a shade different because the first array list the number of tables of the two dimensional arrays. The rest of the two arrays are similar to the twodimensional array in the sense that one is for the row and the other is for the column.

Array Initialization

The initial step of declaring of an array is done. You have given it a variable name and also declared its size but upto this stage the array is empty. It is like a box without any book in it. Similarly declaration of variable is done but some values have to be passed into it. You must be aware of initializing an ordinary variables but how do you initialize an array in which more than one value has to be initialized into one array variable.

Entering Data into an Array :

You have declared an array by giving it a specific name and size you can try and explore the individualities of each element in the array, that is to say how each element is addressed in an array . Array element indexing always start with 0.

Number [0]	1st element
Number [1]	2nd element
Number [2]	3rd element
Number [3]	4th element

This means that [0] will hold the first value, [1] the second, [2] the third & so on. If you want to access the fourth element of the array "sal" you have to write sal [3].

C describes two distinct methods :

a) Static Initialization

b) Dynamic Initialization

Static :

The array once initialized in this cannot be altered by the user. This means that each time the program is executed, the same values are utilized and they can be altered only from the code. It doesn't require the user to input values for the array. This type of initialization is done at the time of the declaration of the array. The values to be inserted are mentioned in the braces `int sal [4] = { 101,102,103,104,105 } ;`
`int a [2] = { 201,202,320 } ;`

In the static initialization the number of values to be allocated is mentioned in the square bracket and the elements are mentioned in the curly braces. The elements are initialized in a static mode.

Dynamic :

In static initialization you noticed that the elements of the array are initialized while declaring the array, but there is a problem. If you want to alter the values or if you want the user to supply the values then static initialization will turn out to be a raw thing. That is where dynamic initialization takes control of the reigns. The user can be allowed to enter the values of his choice into the array and when the program returns the values are not retained and they cover the way for new values. Dynamic initialization can be done using the For Loop `for (i = 0 ; i <= 10 ; i++)`

```
{  
    printf ( "Enter values " );  
    scanf ( "%d", &sal [ i ] );  
}
```

Array Inside the Memory

The elements are stored in a serial order or they are stored randomly .

```
int sal [ 8 ] ;
```

The statement indicates that an array variable is to be created which is of an integer data type and the array can hold 8 elements. In the memory 16 bytes will be reserved because each integer will be 2 Bytes long. The array is initialized with some numbers, this array will host some unrelated numbers. Once the array is initialized the elements are stored in a continuous manner.

```
201 202 203 204 205 206 207 208
```

```
3112 3114 3116 3118 3120 3122 3124 3124
```


The number entered by the user are registered in a successive pattern and each element takes 2 bytes. This is exactly how it represented in the memory of a computer. The number below the element location is the address of the location and each element has its unique address in the memory by which it is identified.

Multi Dimensional Arrays :

Multi dimensional array requires separate brackets for each subscript. One dimensional array has a pair of square brackets, a two dimensional array will have two pairs of square brackets, three dimensional array three pairs of square brackets and so on.

Two Dimensional Array :

A two dimensional array can be visualized as an array below an array. Two dimensional array will have two pairs of square brackets.

Syntax :- datatype arrayname [statement 1] [statement 2]

The two dimensional array is often referred to as a matrix. The statement one and two are the subscripts that the two dimensional array will hold. One will specify the row and expression, two will specify the column and they indicate the number of array elements associated with each subscript.

```
int A [ 4 ] [ 2 ] ;
```

	Column 0	Column 1
Row No 0	4200	10
Row No 1	6500	20
Row No 2	7850 30	
Row No 3	3450 40	

Thus the element in $A[0][0]$ will be 1000 and the element $A[3][3]$ will be 40. The rest of the numbers can be figured by having a look at the above table. In the memory the element are placed in a sequential pattern such that the element $[0][0]$ will be first followed by the element in $[0][1]$ followed by $[1][0]$ and so on.

Character Arrays

C is also flexible that it extends its services to character also. It means that arrays can be formed using even characters. One such array of characters is called string. A string is a collection of characters that are in a particular order. `char name [10] = "Ankush" ;`

In the above example name is an array with an element holding capacity of 10 and it holds a string called "Ankush" The first element is stored in the $[0]$ th position of the array name. That is, the alphabet A will stored in the $[0]$ th position , n will be in the $[1]$ st position and so on till $[9]$ th element.

A doubt may arise that while the array can accommodate 10 values only 6 have been entered. What happens to the rest? Since only 4 values have been accommodated the 7th value will be a null character (`\0`) which represents the end of the string. This will placed automatically at the end of the string if any of the array indexes are vacant. A blank space cannot exist between any two characters while taking input from the user unless accompanied by a specific format specifier.

Strings

A group of integers can be stored in an integer array. Similarly a group of characters can be stored in a character array. Character arrays are many a time also called strings. Most languages internally treat strings as character arrays, but somehow conceal this fact from the programmer. Character arrays or strings are used by programming languages to manipulate text such as words and sentences. A string constant is a one dimensional array of characters terminated by a null (`'\0'`) for example, `char name[] = {'I', 'T', 'T', 'C', 'O', 'M', 'P', '←', 'T', 'E', 'R', '\0'}`

Each character in the array occupies one byte of memory and the last character is always `'\0'`. → That character is this? It looks like two characters, but it is actually only one character, either the `\` indicating that what follows it is something special. `'\0'` is called null character. Note that `'\0'` and `'0'` are not same. ASCII value of `'\0'` is 0 whereas ASCII value of `'0'` is 48. It shows a way a character array is stored in memory. The elements of character array are stored in contiguous memory locations. The terminating null (`'\0'`) is important, because it is the only way the functions that work with a string can know where the string ends. A string not terminated by a `'\0'` is really a string, but merely a collection of characters.

With C compiler a large set of useful string handling library functions are provided.

Function	Use
<code>strlen</code>	Finds length of a string
<code>strlwr</code>	Converts a string to lowercase
<code>strupr</code>	Converts a String to Upper Case
<code>strcat</code>	Appends one string at the end of another
<code>strcpy</code>	Copies a String into Another
<code>strcmp</code>	Compares two strings
<code>strdup</code>	Duplicating a String
<code>strrev</code>	Reversing a String

`strlen ()`

This function counts the number of characters present in a string.

```
void main ( )
{
    char array[] = "Hello ", new[ 20 ];
    int size, size1, size2 ;           printf ( "Enter a
String " );           scanf ( " %s ",new ) ;
    size = strlen ( array ) ;           size1= strlen ("
```

```

SoftBrain Computers " );          size2 = strlen
(new );
printf ( " \n Length of string %s are %d " ,array,size ) ; printf ( " \n Length of Second
string %s are %d " ,"SoftBrain
Computers" ,size1 );

printf ( " \n Length of Third string %s are %d " , new,size2 );
getch ( ) ;
}

```

strcpy ()

This function copies the contents of one string into another. The base addresses of the source and target strings should be supplied to this function.

```

void main ( )
{
char source [ ] =" SoftBrain Computers " ;
char target [20], old [10], new [10] ;
printf ( " Enter a String " ) ; scanf ( " %s " ,
old ) ; strcpy ( target, source ) ; strcpy
( new ,old ) ;
printf ( "\n Source String are %s and target String are %s " , source, target ) ; printf ( "\n
Second Source String are %s and Second target String are %s " , old, new ) ;
getch ( ) ;
}

```

On supplying the base addresses strcpy () goes on copying the characters in source string into target string till it doesn't encounter the end of source string (` \0 `). It is our responsibility to see to it that the target string's dimension is big enough to hold the string meal, character by character .

strcmp ()

This is a function which compares two strings to find out whether they are same or different. The two strings are compared character by character until there is a mismatch or end of one of the strings is reached, which ever occurs first. If the two strings are identical, strcmp () returns a value 0. If they are not, it returns the numeric difference between the ASCII values of the non-matching characters.

Function

A function is a self contained block of a statements that perform a coherent task of some kind. Every C program can be thought of as a collection of these functions. Sometimes the interaction with this person is very simple sometimes it's complex.

You have a task which is always performed exactly in the same way... say a servicing of your motorbike. When you want it to be done, you go to the service station and say " I's time, do it now " → you don't need to give instructions, because the mechanic knows his job. → you don't need to be told when the job is done. ← assume the bike would be serviced in the usual way, the mechanic does.

A simple C function which operated in much the same way as the ,mechanics. We will be looking at two things a program that calls or activates the function and the function itself.

```
void main ( )
{
    message ( ) ;
    printf ( " \n Thanks after return of first function " ) ;
    getch ( ) ;
}
message ( )
{
    printf ( " \n →elcome to the first function program " ) ;
}
}
```

Note :

Any C program contains at least one functions.

If a program contains only one function, it must be **main ()** .

In a C program if there are more than one functions present, then one (and only one) of these functions must be **main()** , because program execution always begins with **main ()** .

There is no limit on the number of functions that might be presented in a C program. Each function in a program is called in the sequence specified by the function calls in **main()** . After each function has done its things, control returns to **main ()** . When **main ()** runs out of function calls, the program ends.

Functions may be divided into three categories.

- Without arguments without return value
- With arguments without return value
- With arguments with return value

Without arguments without return value

For functions of this type neither the caller function nor the called function take any arguments. When a function is called the control jumps directly to the function and executes the statements that are specified within the called function. After the statement is the called function are executed, the control returns to the main program.

With arguments without return value

For function this the caller function transport some values to the called function or the function definitely where these values are used for some specific function or computation but once the computed value is found out within the called function the value is not returned back to the caller function .

With arguments with return value

In this type of functions the values as the arguments are being passed into the called functions where they will be computed and a definite result is relayed back to the caller function and the value will be printed from there.

Advantage of Functions :

A programmer can identify whatever actions are performed repeatedly on various parts of the program, write them as a function and call the function from various part of the program whenever needed. This approach avoids code duplication. Instead of duplicating the same code in different parts of the program, you just write the function once and call it from different parts of the program thus reducing the executable file size and redundancy of code, making the maintenance of code easy.

In a multi tasking system like Unix there is a direct relationship between the amount of memory a program consumes and the speed with which it runs You will in general find that large programs which demand a lot of memory sluggish.

Breaking down your program into subroutines lends your program a structure. You can divide your task into multiple sub-tasks, develop a function for each sub task, and integrated them into a single program. Once a function is written and completely tested, a programmer can expect it to smoothly fit in with the rest of the program modules. Once a function is written, tested and found to be working as per the expectations , the function can be loaded into a library, and can be used in an all together different application. This approaches encourages code reusability.

The types of variables are :

- Local Variables
- Global Variables

Local Variables :

The fact that main is a function and you can declare any number of variables within it. So if a rule holds good for one function why can't it for other functions ? It definitely can.

You can declare a variable inside any function belonging to any size and type. Thus variables declared inside a function are called local variables and are valid and can be used within that function. The life of these variables exists until they are within the function. It is same as the value of a particular variables in a program cannot exist for all the programs. Any attempt to break the rules will result an error.

Global Variables :

Before you can accuse C of being inefficient in certain areas it pops up with an alternative. The alternative for reducing the before mentioned problem is declaring the variables globally. The global variables are otherwise called external variables and unlike the local variables which are confined only to the function which has them, the global variables can be used anywhere in the program The global variables are declared outside the main i.e. before beginning the program and the values hold good through out the program. The

storage for global variables is in a fixed region of memory set aside for this specific purpose by the compiler. The effectiveness of declaring globally can be felt when many functions in a program use the same data.

Defining a variable as global is particularly advantageous because its value is made available to all other functions. But the only risk factor involved in this is that if any function manipulates or alter the global variable then any other function using the same variable at the same time gets the altered value.

Recursion in Functions

Recursion in general terms means repeating itself. A recursive function is a process where the function call itself repeatedly until a particular condition is satisfied. It is more like the loop conditions . But the difference lies in that each action or recursion is based in the result of the previous one.

A recursive result triggers the recursive action for the next result until the condition is satisfied. For a problem to be solved recursively there are two conditions to be satisfied. The problem must be written in a recursive form, The recursion must have a condition which seals the recursion.

C program to check odd or even using modulus operator

```
#include<stdio.h>
```

```
main()
```

```
{ int n;
```

```
    Printf ("Enter an integer\n");
```

```
    Scanf ("%d",&n);
```

```
    if ( n%2 == 0 )  
        Printf ("Even\n");
```

```
    Else
```

```
        Printf ("Odd\n");
```

```
    return 0; }
```

We can use bitwise AND (&) operator to check odd or even, as an example consider binary of 7 (0111) when we perform 7 & 1 the result will be one and you may observe that the least significant bit of every odd number is 1, so (odd number & 1) will be one always and also (even number & 1) is zero.

C program to check odd or even using bitwise operator

```
#include<stdio.h>
```

```

main()
{ int n;

  printf("Enter an integer\n");
  scanf("%d",&n);

  if ( n & 1 == 1 )
  printf("Odd\n"); else
  printf("Even\n");

  return 0; }

```

Find odd or even using conditional operator

```

#include<stdio.h>

main()
{ int n;

  printf("Input an integer\n");
  scanf("%d",&n);

  n%2 == 0 ? printf("Even\n") : printf("Odd\n");

  return 0; }

```

C program to check odd or even without using bitwise or modulus operator

```

#include<stdio.h>

main()
{ int n;

  printf("Enter an integer\n");
  scanf("%d",&n);

  if ( (n/2)*2 == n )
  printf("Even\n"); else
  printf("Odd\n");

  return 0; }

```

In c programming language when we divide two integers we get an integer result, For example the result of 7/3 will be 2. So we can take advantage of this and may use it to

find whether the number is odd or even. Consider an integer n we can first divide by 2 and then multiply it by 2 if the result is the original number then the number is even otherwise the number is odd. For example $11/2 = 5$, $5*2 = 10$ (which is not equal to eleven), now consider $12/2 = 6$ and $6 *2 = 12$ (same as original number). These are some logic which may help you in finding if a number is odd or not.

Prime number program in c language

```
#include<stdio.h>

int main()
{
    int n, i = 3, count, c;

    printf("Enter the number of prime numbers required\n");
    scanf("%d",&n);

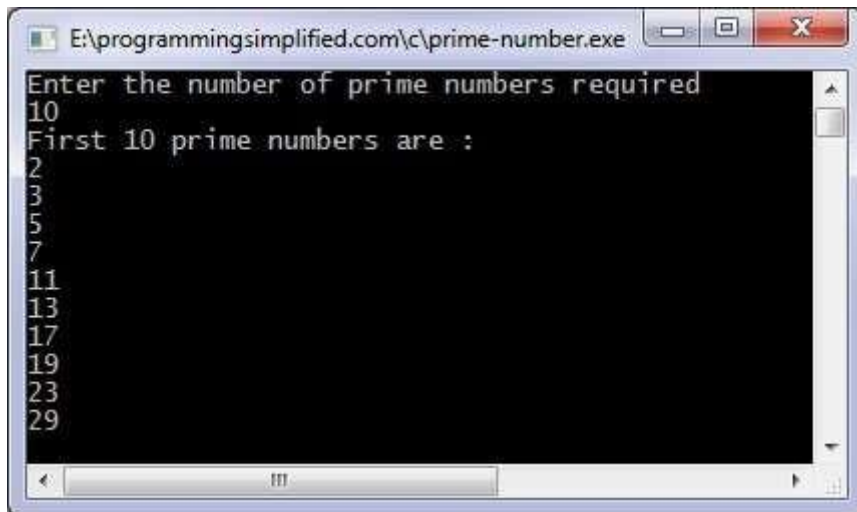
    if ( n >= 1 )
    {
        printf("First %d prime numbers are :\n",n);
        printf("2\n");
    }

    for ( count = 2 ; count <= n ; )
    {
        for ( c = 2 ; c <= i - 1 ; c++ )
        {
            if ( i%c == 0 )
                break;
        }
        if ( c == i )
        {
            printf("%d\n",i);
            count++;
        }
        i++;
    }

    return 0; }
```

Download [Prime number](#) program.

Output of program:



```
E:\programmingsimplified.com\c\prime-number.exe
Enter the number of prime numbers required
10
First 10 prime numbers are :
2
3
5
7
11
13
17
19
23
29
```

C program for prime number or not

```
#include<stdio.h>

main()
{
    int n, c = 2;

    printf("Enter a number to check if it is prime\n"); scanf("%d",&n);

    for ( c = 2 ; c <= n - 1 ; c++ )
    {
        if ( n%c == 0 )
        {
            printf("%d is not prime.\n", n);
            break;
        } } if
(c == n)
    printf("%d is prime.\n", n);

    return 0; }
```

C program for prime number using function

```
#include<stdio.h>

int check_prime(int);

main()
{
    int n, result;
```

```

printf("Enter an integer to check whether it is prime or not.\n");  scanf("%d",&n);

result = check_prime(n);

if ( result == 1 )
    printf("%d is prime.\n", n);
else
    printf("%d is not prime.\n", n);

return 0;
}

int check_prime(int a)
{ int c;

for ( c = 2 ; c <= a - 1 ; c++ )
{
    if ( a%c == 0 )
        return 0;
} if ( c ==
a ) return
1; }

```

There are many logic to check prime numbers, one given below is more efficient than above method.

```
for ( c = 2 ; c <= (int)sqrt(n) ; c++ )
```

Only checking from 2 to square root of number is sufficient. There are many more efficient logic available.

PROGRAMMING LANGUAGE C++

Introduction

object-orientation is introduced as a new programming concept which should help you in developing high quality software. Some people will say that object-orientation is ``modern". When reading announcements of new products everything seems to be ``object-oriented". ``Objects" are everywhere. In this section we will try to outline characteristics of object-orientation to allow you to judge those object-oriented products. Roughly speaking, we can distinguish the learning curve of someone who learns to program:

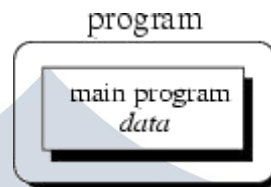
- Unstructured programming,
- procedural programming,

- modular programming and □ object-oriented programming.

Unstructured Programming

Usually, people start learning programming by writing small and simple programs consisting only of one main program. Here "main program" stands for a sequence of commands or *statements* which modify data which is *global* throughout the whole program.

Figure: Unstructured programming. The main program directly operates on global data.



Procedural Programming

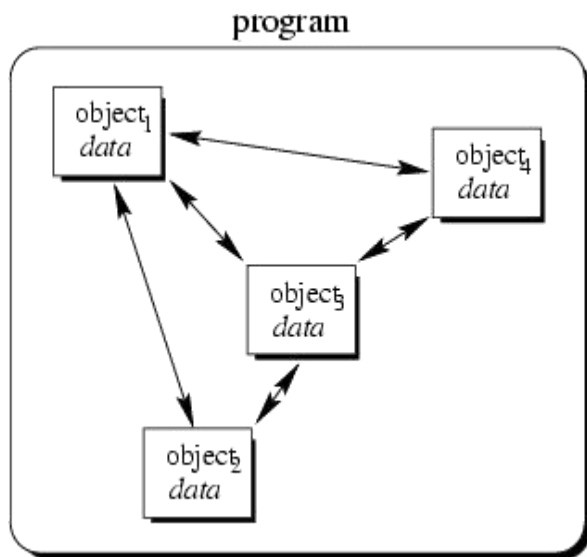
With procedural programming you are able to combine returning sequences of statements into one single place. A *procedure call* is used to invoke the procedure. After the sequence is processed, flow of control proceeds right after the position where the call was made

Modular Programming

With modular programming procedures of a common functionality are grouped together into separate *modules*. A program therefore no longer consists of only one single part. It is now divided into several smaller parts which interact through procedure calls and which form the whole program.

Object-oriented programming solves some of the problems just mentioned. In contrast to the other techniques, we now have a web of interacting objects, each house-keeping its own state.

Figure: Object-oriented programming. Objects of the program interact by sending messages to each other.



Abstract Data Types

Some authors describe object-oriented programming as programming *abstract data types* and their relationships. ADTs are used to define a new type from which instances can be created. ADTs provide an abstract view to describe properties of sets of entities, their use is independent from a particular programming language.

Class

A class is an actual representation of an ADT. It therefore provides implementation details for the data structure used and operations. We play with the ADT Integer and design our own class for it:

```

class Integer
{
    attributes:
    inti

    methods:
    setValue(int n)
    Integer addValue(Integer j)
}

```

A class is the implementation of an abstract data type (ADT).

*It defines attributes and methods which implement the data structure and operations of the ADT, respectively. Instances of classes are called *objects*. Consequently, classes define properties and behaviour of sets of objects.*

Object

Objects are uniquely identifiable by a *name*. Therefore you could have two distinguishable objects with the same set of values. This is similar to "traditional" programming languages where you could have, say two integers *i* and *j* both of which equal to "2". Please notice the use of "i" and "j" in the last sentence to name the two integers. We refer to the set of values at a particular time as the *state* of the object.

Definition (Object) An object is an instance of a class. It can be uniquely identified by its name and it defines a state which is represented by the values of its attributes at a particular time. The **behaviour** of an object is defined by the set of methods which can be applied on it.

Message

A running program is a pool of objects where objects are created, destroyed and *interacting*. This interacting is based on *messages* which are sent from one object to another asking the recipient to apply a method on itself. A **message** is a request to an object to invoke one of its methods. A message therefore contains

- the **name** of the method and
- the **arguments** of the method.

Inheritance

With inheritance we are able to make use of the a-kind-of and is-a relationship. As described there, classes which are a-kind-of another class share properties of the latter. In our point and circle example, we can define a circle which *inherits from* point:

```
class Circle inherits from Point
{attributes: int radius
```

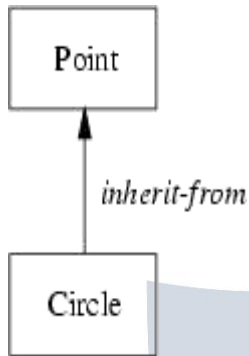
```
methods:
  setRadius(int          newRadius)
  getRadius()}}
```

Class *Circle* inherits all data elements and methods from point. There is no need to define them twice: We just use already existing and well-known data and method definitions.

Definition (Inheritance) Inheritance is the mechanism which allows a class A to inherit properties of a class B. We say "A inherits from B". Objects of class A thus have access to attributes and methods of class B without the need to redefine them. The following definition defines two terms with which we are able to refer to participating classes when they use inheritance.

Definition (Superclass/Subclass) If class A inherits from class B, then B is called superclass of A. A is called subclass of B. Objects of a subclass can be used where objects of the corresponding superclass are expected. This is due to the fact that objects of the subclass share the same behaviour as objects of the superclass.

Figure: A simple inheritance graph.

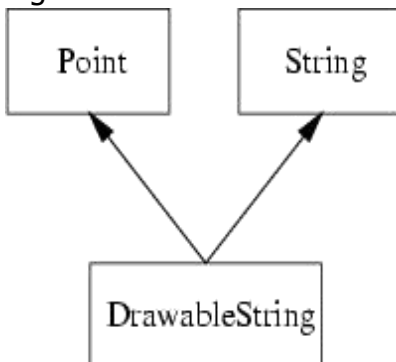


Multiple Inheritance

One important object-oriented mechanism is multiple inheritance. Multiple inheritance does **not** mean that multiple subclasses share the same superclass. It also does **not** mean that a subclass can inherit from a class which itself is a subclass of another class. Multiple inheritance means that one subclass can have *more than one* superclass. This enables the subclass to inherit properties of more than one superclass and to "merge" their properties.



Figure: Derive a drawable string which inherits properties of Point and String.



Definition (Multiple Inheritance) If class *A* inherits from more than one class, ie. *A* inherits from *B*₁, *B*₂, ..., *B*_{*n*}, we speak of **multiple inheritance**. This may introduce **naming conflicts** in *A* if at least two of its superclasses define properties with the same name.

Inheritance between classes

A key feature of C++ classes is inheritance. Inheritance allows to create classes which are derived from other classes, so that they automatically include some of its "parent's" members, plus its own. For example, we are going to suppose that we want to declare a series of classes that describe polygons like our CRectangle, or like CTriangle. They have certain common properties, such as both can be described by means of only two sides: height and base.

This could be represented in the world of classes with a class CPolygon from which we would derive the two other ones: CRectangle and CTriangle.

What is inherited from the base class?

In principle, a derived class inherits every member of a base class except:

- its constructor and its destructor
- its operator=() members
- its friends

Although the constructors and destructors of the base class are not inherited themselves, its default constructor (i.e., its constructor with no parameters) and its destructor are always called when a new object of a derived class is created or destroyed.

Abstract Classes

With inheritance we are able to force a subclass to offer the same properties like their superclasses. Consequently, objects of a subclass behave like objects of their superclasses. Sometimes it make sense to only describe the properties of a set of objects without knowing the actual behaviour beforehand.

Static and Dynamic Binding

In strongly typed programming languages you typically have to *declare* variables prior to their use. This also implies the variable's *definition* where the compiler reserves space for the variable. **Definition (Static Binding)** If the type *T* of a variable is explicitly associated with its name *N* by declaration, we say, that *N* is statically bound to *T*. The association process is called *static binding*.

Polymorphism

Polymorphism allows an entity (for example, variable, function or object) to take a variety of representations. Therefore we have to distinguish different types of polymorphism which will be outlined here.

The first type is similar to the concept of dynamic binding. Here, the type of a variable depends on its content. Thus, its type depends on the content at a specific time:

```
v := 123      /* v is integer */ ...  
/* use v as integer */ v := 'abc' /* v  
"switches" to string */ ... /* use  
v as string */
```

Definition (Polymorphism) The concept of dynamic binding allows a variable to take different types dependent on the content at a particular time. This ability of a variable is called polymorphism. Another type of polymorphism can be defined for functions. For example, suppose you want to define a function isNull() which returns TRUE if its argument is 0 (zero) and FALSE otherwise.

Function templates

Function templates are special functions that can operate with generic types. This allows us to create a function template whose functionality can be adapted to more than one type or class without repeating the entire code for each type. In C++ this can be achieved using template parameters. A template parameter is a special kind of parameter that can be used to pass a type as argument: just like regular function parameters can be used to pass values to a function, template parameters allow to pass also types to a function. These function templates can use these parameters as if they were any other regular type.

The format for declaring function templates with type parameters is: `template <class identifier> function_declaration;` `template <typename identifier> function_declaration;`

The only difference between both prototypes is the use of either the keyword `class` or the keyword `typename`.

Template specialization

If we want to define a different implementation for a template when a specific type is passed as template parameter, we can declare a specialization of that template. For example, let's suppose that we have a very simple class called `mycontainer` that can store one element of any type and that it has just one member function called `increase`, which increases its value. But we find that when it stores an element of type `char` it would be more convenient to have a completely different implementation with a function member `uppercase`, so we decide to declare a class template specialization for that type.

Exceptions

Exceptions provide a way to react to exceptional circumstances (like runtime errors) in our program by transferring control to special functions called handlers. To catch exceptions we must place a portion of code under exception inspection. This is done by enclosing that portion of code in a try block. When an exceptional circumstance arises within that block, an exception is thrown that transfers the control to the exception handler. If no exception is thrown, the code continues normally and all handlers are ignored.

Standard exceptions

The C++ Standard library provides a base class specifically designed to declare objects to be thrown as exceptions. It is called `exception` and is defined in the `<exception>` header file under the namespace `std`. This class has the usual default and copy constructors, operators and destructors, plus an additional virtual member function called `what` that returns a null-terminated character sequence (`char *`) and that can be overwritten in derived classes to contain some sort of description of the exception.

Type Casting

Converting an expression of a given type into another type is known as type-casting. We have already seen some ways to type cast:

Implicit conversion

Implicit conversions do not require any operator. They are automatically performed when a value is copied to a compatible type.

Explicit conversion

C++ is a strong-typed language. Many conversions, specially those that imply a different interpretation of the value, require an explicit conversion.

Data Types

Table: Built-in types.

Type	Description	Size	Domain
char	Signed character/byte. Characters are enclosed in single quotes.	1	-128..127
double	Double precision number	8	ca. $10^{-308}..10^{308}$
int	Signed integer	4	$-2^{31}..2^{31} - 1$
float	Floating point number	4	ca. $10^{-38}..10^{38}$
long (int)	Signed long integer	4	$-2^{31}..2^{31} - 1$
long long (int)	Signed very long integer	8	$-2^{63}..2^{63} - 1$
short (int)	Short integer	2	$-2^{15}..2^{15} - 1$
unsigned char	Unsigned character/byte	1	0..255
unsigned (int)	Unsigned integer	4	$0..2^{32} - 1$
unsigned long (int)	Unsigned long integer	4	$0..2^{32} - 1$
unsigned long long (int)	Unsigned very long integer	8	$0..2^{64} - 1$
unsigned short (int)	Unsigned short integer	2	$0..2^{16} - 1$

Expressions and Operators

Expressions are combined of both *terms* and *operators*. The first could be constants, variables or expressions. From the latter, C offers all operators known from other languages. However, it offers some operators which could be viewed as abbreviations to combinations of other operators. In C almost everything is an expression. For example, the assignment statement `ix = 12;` returns the value of its righthand operand. As a "side effect" it also sets the value of the lefthand operand. Thus, `ix = 12;`

sets the value of *ix* to 12 (assuming that *ix* has an appropriate type). Now that the assignment is also an expression, we can combine several of them; for example:

```
kx = jx = ix = 12;
```

What happens? The first assignment assigns *kx* the value of its righthand side. This is the value of the assignment to *jx*. But this is the value of the assignment to *ix*. The value of this latter is 12 which is returned to *jx* which is returned to *kx*. Thus we have expressed

```
ix = 12;  
jx = 12;  
kx = 12; in  
one line.
```

Statements

C defines all usual flow control statements. Statements are terminated by a semicolon ` `";. We can group multiple statements into blocks by enclosing them in curly brackets. Within each block, we can define new variables:

```
{  
  int i;      /* Define a global i */  i  
  = 1;      /* Assign i the value 0 */  {  
  /* Begin new block */  int i; /*  
  Define a local i */  i = 2; /* Set  
  its value to 2 */  
  }          /* Close block */  
            /* Here i is again 1 from the outer block */  
}
```

Table of Statements

VIDYAPITH ACADEMY

WWW.VIDYAPITH.IN

- ESTD 2020 -

Statement	Description
<code>break;</code>	Leave current block. Also used to leave case statement in switch .
<code>continue;</code>	Only used in loops to continue with next loop immediately.
<code>do stmt while (expr);</code>	Execute <i>stmt</i> as long as <i>expr</i> is TRUE.
<code>for ([expr]; [expr]; [expr]) stmt</code>	This is an abbreviation for a while loop where the first <i>expr</i> is the initialization, the second <i>expr</i> is the condition and the third <i>expr</i> is the step.
<code>goto label;</code>	Jumps to position indicated by <i>label</i> . The destination is <i>label</i> followed by colon ":".
<code>if (expr) stmt [else stmt]</code>	IF-THEN-ELSE in C notation
<code>return [expr];</code>	Return from function. If function returns void return should be used without additional argument. Otherwise the value of <i>expr</i> is returned.
<code>switch (expr) { case const-expr: stmts case const-expr: stmts ... [default: stmts] }</code>	After evaluation of <i>expr</i> its value is compared with the case clauses. Execution continues at the one that matches. BEWARE: You must use break to leave the switch if you don't want execution of following case clauses! If no case clause matches and a default clause exists, the statements of the default clause are executed.
<code>while (expr) stmt</code>	Repeat <i>stmt</i> as long as <i>expr</i> is TRUE.

Functions

As C is a procedural language it allows the definition of *functions*. Procedures are "simulated" by functions returning "no value". This value is a special type called void.

Functions are declared similar to variables, but they enclose their arguments in parenthesis (even if there are no arguments, the parenthesis must be specified):

```
int sum(int to); /* Declaration of sum with one argument */
int bar();      /* Declaration of bar with no arguments */ void
foo(int ix, int jx);
                /* Declaration of foo with two arguments */
```

Pointers and Arrays

One of the most common problems in programming in C++ is the understanding of pointers and arrays. In C (C++) both are highly related with some small but essential differences. You declare a pointer by putting an asterisk between the data type and the name of the variable or function:

```
char *strp;    /* strp is `pointer to char' */
```

You access the content of a pointer by dereferencing it using again the asterisk:

```
*strp = 'a';   /* A single character */
```

As in other languages, you must provide some space for the value to which the pointer points. A pointer to characters can be used to point to a sequence of characters: the *string*. Strings in C are terminated by a special character NUL (0 or as char '\0').

Arrays as parameters

At some moment we may need to pass an array to a function as a parameter. In C++ it is not possible to pass a complete block of memory by value as a parameter to a function, but we are allowed to pass its address. In practice this has almost the same effect and it is a much faster and more efficient operation.

Pointers

We have already seen how variables are seen as memory cells that can be accessed using their identifiers. This way we did not have to care about the physical location of our data within memory, we simply used its identifier whenever we wanted to refer to our variable. The memory of your computer can be imagined as a succession of memory cells, each one of the minimal size that computers manage (one byte). These single-byte memory cells are numbered in a consecutive way, so as, within any block of memory, every cell has the same number as the previous one plus one. This way, each cell can be easily located in the memory because it has a unique address and all the memory cells follow a successive pattern. For example, if we are looking for cell 1776 we know that it is going to be right between cells 1775 and 1777, exactly one thousand cells after 776 and exactly one thousand cells before cell 2776.

Reference operator (&)

As soon as we declare a variable, the amount of memory needed is assigned for it at a specific location in memory (its memory address). We generally do not actively decide the exact location of the variable within the panel of cells that we have imagined the memory to be - Fortunately, that is a task automatically performed by the operating system during runtime. However, in some cases we may be interested in knowing the address where our variable is being stored during runtime in order to operate with relative positions to it.

Dereference operator (*)

We have just seen that a variable which stores a reference to another variable is called a pointer. Pointers are said to "point to" the variable whose reference they store. Using a pointer we can directly access the value stored in the variable which it points to. To do this, we simply have to precede the pointer's identifier with an asterisk (*), which acts as dereference operator and that can be literally translated to "value pointed by".

Null pointer

A null pointer is a regular pointer of any pointer type which has a special value that indicates that it is not pointing to any valid reference or memory address. This value is the result of type-casting the integer value zero to any pointer type.

```
int * p; p = 0; // p has a null pointer
```

value

Do not confuse null pointers with void pointers. A null pointer is a value that any pointer may take to represent that it is pointing to "nowhere", while a void pointer is a special type of pointer that can point to somewhere without a specific type. One refers to the value stored in the pointer itself and the other to the type of data it points to.

Pointers to functions

C++ allows operations with pointers to functions. The typical use of this is for passing a function as an argument to another function, since these cannot be passed dereferenced. In order to declare a pointer to a function we have to declare it like the prototype of the function except that the name of the function is enclosed between parentheses () and an asterisk (*) is inserted before the name:

C++ Program

Here we introduce the first program which is so often used: a program which prints ``Hello, world!" to your screen:

```
#include <stdio.h>
```

```

/* Global variables should be here */

/* Function definitions should be here */
int
main() {
    puts("Hello,      world!");
return 0;
} /* main */

```

The first line looks something strange. Its explanation requires some information about how C (and C++) programs are handled by the compiler. The compilation step is roughly divided into two steps. The first step is called "preprocessing" and is used to prepare raw C code. In this case this step takes the first line as an argument to include a file called *stdio.h* into the source. The angle brackets just indicate, that the file is to be searched in the standard search path configured for your compiler. The file itself provides some declarations and definitions for standard input/output. For example, it declares a function called *put()*. The preprocessing step also deletes the comments.

Constructor

With constructors we are able to initialize our objects at definition time as we have requested it for our singly linked list. We are now able to define a class *List* where the constructors take care of correctly initializing its objects. If we want to create a point from another point, hence, copying the properties of one object to a newly created one, we sometimes have to take care of the copy process. For example, consider the class *List* which allocates dynamically memory for its elements. If we want to create a second list which is a copy of the first, we must allocate memory and copy the individual elements.

Destructors

Consider a class *List*. Elements of the list are dynamically appended and removed. The constructor helps us in creating an initial empty list. However, when we leave the scope of the definition of a list object, we must ensure that the allocated memory is released. We therefore define a special method called destructor which is called once for each object at its destruction time:

```

void foo() {
    List alist;    // List::List() initializes to
// empty list.
    ...           // add/remove elements
}                // Destructor call!

```

Destruction of objects take place when the object leaves its scope of definition or is explicitly destroyed. The latter happens, when we dynamically allocate an object and release it when

it is no longer needed. Destructors are declared similar to constructors. Thus, they also use the name prefixed by a tilde (~) of the defining class.

Identifiers

A valid identifier is a sequence of one or more letters, digits or underscore characters (_). Neither spaces nor punctuation marks or symbols can be part of an identifier. Only letters, digits and single underscore characters are valid. In addition, variable identifiers always have to begin with a letter. They can also begin with an underline character (_), but in some cases these may be reserved for compiler specific keywords or external identifiers, as well as identifiers containing two successive underscore characters anywhere. In no case they can begin with a digit. Another rule that you have to consider when inventing your own identifiers is that they cannot match any keyword of the C++ language nor your compiler's specific ones, which are reserved keywords. The standard reserved keywords are:

asm, auto, bool, break, case, catch, char, class, const, const_cast, continue, default, delete,

do, double, dynamic_cast, else, enum, explicit, export, extern, false, float, for, friend, goto, if, inline, int, long, mutable, namespace, new, operator, private, protected, public, register, reinterpret_cast, return, short, signed, sizeof, static, static_cast, struct, switch, template, this, throw, true, try, typedef, typeid, typename, union, unsigned, using, virtual, void, volatile, wchar_t, while.

Operators in C++

Operators can briefly be defined as the tools used for solving various mathematical, conditional, relational and logical problems. The operators that you will be dealing with are arithmetic operators, unary operators, relational and logical operators. The item in between which the operators are placed are called operands.

Arithmetic operators :

Arithmetic Operators can be considered the main point of all the operators. Arithmetic operators are tools that help us in computing various mathematical operations.

Operators	Function Performed
+	Addition
-	Subtraction
*	Multiplication

/	Division
%	(Modulus) Finds the remainder

Unary Operators :

There are two types of unary operators :-

- **The increment operator (denotes as ' ++ ')**
- **The decrement operator (denotes as ' -- ')**

When the increment operator is prefixed to a variable which holds an integer it increases the value of the number by one. Similarly the decrement operator decreases the value by one.

Relational and Logical Operators :

Operators	Significance
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
Equality Operators	
==	Equal to
!=	Not equal to
Logical Operators	
&&	And
	Or
!	Not

Assignment is = *i.e.* i = 4; ch = 'y';

Increment ++, Decrement -- which are more efficient than their long hand equivalents, for example:-- x++ is faster than x=x+1.

The ++ and -- operators can be either in post-fixed or pre-fixed. With pre-fixed the value is computed before the expression is evaluated whereas with post-fixed the value is computed after the expression is evaluated.

Comparison Operators

To test for equality is ==

A warning: Beware of using ``=' instead of ``==', such as writing accidentally

```
if ( i = j ) .....
```

This is a perfectly **LEGAL** C statement (syntactically speaking) which copies the value in "j" into "i", and delivers this value, which will then be interpreted as TRUE if j is nonzero. This is called **assignment by value** -- a key feature of C.

Not equals is: !=

Other operators < (less than) , > (grater than), <= (less than or equals), >= (greater than or equals) are as usual.

Statements in C Language

C needs some kind of managerial system which instructs the computer what function has to be performed at which time. The control statement act as a manager. You will amazed at the functions the different control statements handle. For example in a program you may have a statement in which a multiplication operation is to be performed and in the next an addition operation and after the next line there may be a statement which may want itself to be repeated 8 times and another statement may like itself to be skipped and attended to at a later stage. All these conditions are managed by the control statements.

The sequence control structure is one of the basic uncomplicated control statements. The idea is very simple. In the sequence control instructions the statements to be executed are done so in a serial manner one after the other. The C compiler first executes the opening line and then moves on to the next and so on. In a serial fashion and the entire program is executed in this manner.

Conditional Statement

C program executes program sequentially. Sometimes, a program requires checking of certain conditions in program execution. C provides various key condition statements to check condition and execute statements according conditional criteria. These statements are called as '**Decision Making Statements**' or '**Conditional Statements**'. Followings are the different conditional statements used in C :

5. If Statement
6. If-Else Statement
7. Nested If-Else Statement
8. Switch Case

If Statement:The if statement can be used to test conditions so that we can alter the flow of a program. Code:

```
#include<stdio.h>
int main()
{
    int mark; char pass;
    scanf("%d",&mark);
    if (mark > 40) pass
    = "y"; return 0;
}
```

If-Else Statement:The if statement first tests if a condition is true and then executes an instruction and the else is for when the result of the condition is false. Code:

```
#include<stdio.h> int
main()
{
    int mark; char pass;
    scanf("%d",&mark);
    if (mark > 40)
    {
        pass = "y";
        printf("You passed");
    }
    else
    {
        pass = "n"; printf("You
        failed");
    }
    return 0;
}
```

The Switch Statement:The switch statement is just like an if statement but it has many conditions and the commands for those conditions in only 1 statement. It is runs faster than an if statement. In a switch statement you first choose the variable to be tested and then you give each of the conditions and the commands for the conditions. You can also put in a default if none of the conditions are equal to the value of the variable.

For Loop, While Loop, Break and Continue

Control structures are basically of three types –

- **Sequence statements**
- **Iterative statements**
- **Selection statements**

Sequence Statements : All the State in a program except the iterative & statements. They are generally the individual statements which performs the task of input, output, assignment declaration etc.

Iterative Statement are those repeated execution of a particular set of instructions desired number of times. These statements are generally called loops for their execution nature.

Types of Looping Statements:

Basically, the types of looping statements depends on the condition checking mode. Condition checking can be made in two ways as : Before loop and after loop. So, there are 2(two) types of looping statements.

- **Entry controlled loop**
- **Exit controlled loop**

2. Entry controlled loop :

In such type of loop, the test condition is checked first before the loop is executed.

Some common examples of this looping statements are :

- **while loop**
- **for loop**

2. Exit controlled loop :

In such type of loop, the loop is executed first. Then condition is checked after block of statements are executed. The loop executed atleast one time compulsarily.

Some common example of this looping statement is :

- **do-while loop**

In C language the iterative statements (loops) can be implemented in the three loops and they are

The For Loop

Syntax-

for (initialization; condition; incrementation)

```
{  
    _____body of loop _____  
}
```

For Loop will perform its execution until the condition remains satisfied. If the body of the loop consists of more than one statement then these statements are made compound by placing the open and closed curly brackets around the body of the loop. For loop is a count loop. The initialization condition and incrementation may be done in the same statement. For loop will not execute at least once also if the condition is false at the first time itself.

The While Loop

Syntax –

Initialization;

While (condition)

```
{  
    Body ★ of loop;
```

Incrementation;

```
}
```

In this loop, initialisation, condition and incrementation is done in the three different statements. This loops is count as well as event loop. In case of while loops the body of the loop will consist of more than one statements because each time one statement will be of incrementation. Hence the open and closed curly brackets are required.

Do-While Loop Statement:

The third loop statement available in C is do-while statement syntax :- Initialization;

Do

```
{
```

Body of loop;

```
Incrementation;
```

}while(condition)

Overloaded functions.

In C++ two different functions can have the same name if their parameter types or number are different. That means that you can give the same name to more than one function if they have either a different number of parameters or different types in their parameters.

inline functions.

The inline specifier indicates the compiler that inline substitution is preferred to the usual function call mechanism for a specific function. This does not change the behavior of a function itself, but is used to suggest to the compiler that the code generated by the function body is inserted at each point the function is called, instead of being inserted only once and perform a regular call to it, which generally involves some additional overhead in running time.

Recursivity

Recursivity is the property that functions have to be called by themselves. It is useful for many tasks, like sorting or calculate the factorial of numbers. For example, to obtain the factorial of a number (n!) the mathematical formula would be:

$$n! = n * (n-1) * (n-2) * (n-3) \dots * 1$$

Classes defined with struct and union

Classes can be defined not only with keyword class, but also with keywords struct and union. The concepts of class and data structure are so similar that both keywords (struct and class) can be used in C++ to declare classes (i.e. structs can also have function members in C++, not only data members). The only difference between both is that members of classes declared with the keyword struct have public access by default, while members of classes declared with the keyword class have private access. For all other purposes both keywords are equivalent.

The concept of unions is different from that of classes declared with struct and class, since unions only store one data member at a time, but nevertheless they are also classes and can thus also hold function members. The default access in union classes is public.

Friend functions

In principle, private and protected members of a class cannot be accessed from outside the same class in which they are declared. However, this rule does not affect friends. Friends are functions or classes declared as such. If we want to declare an external function as friend of a class, thus allowing this function to have access to the private and protected members of this class, we do it by declaring a prototype of this external function within the class, and preceding it with the keyword friend.

Preprocessor directives

Preprocessor directives are lines included in the code of our programs that are not program statements but directives for the preprocessor. These lines are always preceded by a hash sign (#). The preprocessor is executed before the actual compilation of code begins, therefore the preprocessor digests all these directives before any code is generated by the statements.

These preprocessor directives extend only across a single line of code. As soon as a newline character is found, the preprocessor directive is considered to end. No semicolon (;) is expected at the end of a preprocessor directive. The only way a preprocessor directive can extend through more than one line is by preceding the newline character at the end of the line by a backslash (\).

macro definitions (#define, #undef)

To define preprocessor macros we can use #define. Its format is:

```
#define identifier replacement
```

When the preprocessor encounters this directive, it replaces any occurrence of identifier in the rest of the code by replacement. This replacement can be an expression, a statement, a block or simply anything. The preprocessor does not understand C++, it simply replaces any occurrence of identifier by replacement.

Input/Output with files

C++ provides the following classes to perform output and input of characters to/from files:

- ofstream: Stream class to write on files
- ifstream: Stream class to read from files
- fstream: Stream class to both read and write from/to files.

These classes are derived directly or indirectly from the classes istream, and ostream. We have already used objects whose types were these classes: cin is an object of class istream and cout is an object of class ostream. Therefore, we have already been using classes that are related to our file streams. And in fact, we can use our file streams the same way we are already used to use cin and cout, with the only difference that we have to associate these streams with physical files.

Open a file

The first operation generally performed on an object of one of these classes is to associate it to a real file. This procedure is known as to open a file. An open file is represented within a program by a stream object (an instantiation of one of these classes, in the previous example this was myfile) and any input or output operation performed on this stream object will be applied to the physical file associated to it. In order to open a file with a stream object we use its member function open():

```
open (filename, mode);
```

Closing a file

When we are finished with our input and output operations on a file we shall close it so that its resources become available again. In order to do that we have to call the stream's member function close(). This member function takes no parameters, and what it does is to flush the associated buffers and close the file:

```
myfile.close();
```

Once this member function is called, the stream object can be used to open another file, and the file is available again to be opened by other processes. In case that an object is destructed while still associated with an open file, the destructor automatically calls the member function close().

Text files

Text file streams are those where we do not include the ios::binary flag in their opening mode. These files are designed to store text and thus all values that we input or output from/to them can suffer some formatting transformations, which do not necessarily correspond to their literal binary value.

Checking state flags

In addition to eof(), which checks if the end of file has been reached, other member functions exist to check the state of a stream (all of them return a bool value):

bad()

Returns true if a reading or writing operation fails. For example in the case that we try to write to a file that is not open for writing or if the device where we try to write has no space left.

fail()

Returns true in the same cases as bad(), but also in the case that a format error happens, like when an alphabetical character is extracted when we are trying to read an integer number.

eof()

Returns true if a file open for reading has reached the end.

good()

It is the most generic state flag: it returns false in the same cases in which calling any of the previous functions would return true.

In order to reset the state flags checked by any of these member functions we have just seen we can use the member function `clear()`, which takes no parameters.

get and put stream pointers

All i/o streams objects have, at least, one internal stream pointer:

`ifstream`, like `istream`, has a pointer known as the get pointer that points to the element to be read in the next input operation. `ofstream`, like `ostream`, has a pointer known as the put pointer that points to the location where the next element has to be written. Finally, `fstream`, inherits both, the get and the put pointers, from `iostream` (which is itself derived from both `istream` and `ostream`).

These internal stream pointers that point to the reading or writing locations within a stream can be manipulated using the following member functions:

tellg() and tellp()

These two member functions have no parameters and return a value of the member type `pos_type`, which is an integer data type representing the current position of the get stream pointer (in the case of `tellg`) or the put stream pointer (in the case of `tellp`).

seekg() and seekp()

These functions allow us to change the position of the get and put stream pointers. Both functions are overloaded with two different prototypes. The first prototype is:

- `seekg (position);`
- `seekp (position);`

Using this prototype the stream pointer is changed to the absolute position position (counting from the beginning of the file). The type for this parameter is the same as the one returned by functions `tellg` and `tellp`: the member type `pos_type`, which is an integer value

Binary files

In binary files, to input and output data with the extraction and insertion operators (`<<` and `>>`) and functions like `getline` is not efficient, since we do not need to format any data, and

data may not use the separation codes used by text files to separate elements (like space, newline, etc...). File streams include two member functions specifically designed to input and output binary data sequentially: write and read. The first one (write) is a member function of ostream inherited by ofstream. And read is a member function of istream that is inherited by ifstream. Objects of class fstream have both members.

Buffers and Synchronization

When we operate with file streams, these are associated to an internal buffer of type streambuf. This buffer is a memory block that acts as an intermediary between the stream and the physical file. For example, with an ofstream, each time the member function put (which writes a single character) is called, the character is not written directly to the physical file with which the stream is associated. Instead of that, the character is inserted in that stream's intermediate buffer. When the buffer is flushed, all the data contained in it is written to the physical medium (if it is an output stream) or simply freed (if it is an input stream). This process is called synchronization and takes place under any of the following circumstances:

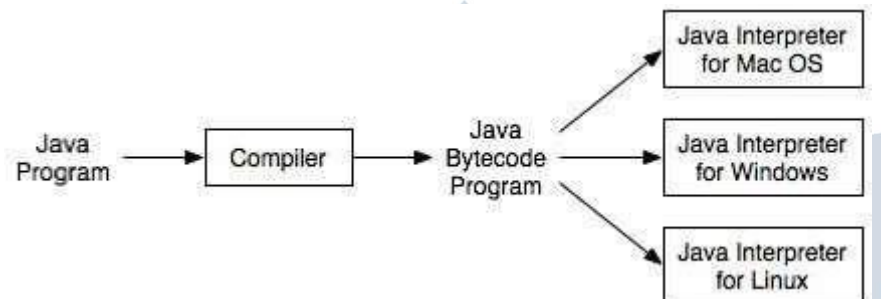
- When the file is closed: before closing a file all buffers that have not yet been flushed are synchronized and all pending data is written or read to the physical medium.
- When the buffer is full: Buffers have a certain size. When the buffer is full it is automatically synchronized.
- Explicitly, with manipulators: When certain manipulators are used on streams, an explicit synchronization takes place. These manipulators are: flush and endl.
- Explicitly, with member function sync(): Calling stream's member function sync(), which takes no parameters, causes an immediate synchronization. This function returns an int value equal to -1 if the stream has no associated buffer or in case of failure. Otherwise (if the stream buffer was successfully synchronized) it returns 0.

PROGRAMMING IN JAVA

MACHINE LANGUAGE CONSISTS of very simple instructions that can be executed directly by the CPU of a computer. The designers of Java chose to use a combination of compilation and interpretation. Programs written in Java are compiled into machine language, but it is a machine language for a computer that doesn't really exist. This so-called "virtual" computer is known as the Java Virtual Machine, or JVM. The machine language for the Java Virtual Machine is called Java bytecode. There is no reason why Java bytecode couldn't be used as the machine language of a real computer, rather than

a virtual computer. But in fact the use of a virtual machine makes possible one of the main selling points of Java: the fact that it can actually be used on any computer. All that the computer needs is an interpreter for Java bytecode. Such an interpreter simulates the JVM in the same way that Virtual PC simulates a PC computer. (The term JVM is also used for the Java bytecode interpreter program that does the simulation, so we say that a computer needs a JVM in order to run Java programs. Technically, it would be more correct to say that the interpreter implements the JVM than to say that it is a JVM.)

Of course, a different Java bytecode interpreter is needed for each type of computer, but once a computer has a Java bytecode interpreter, it can run any Java bytecode program. And the same Java bytecode program can be run on any computer that has such an interpreter. This is one of the essential features of Java: the same compiled program can be run on many different types of computers.



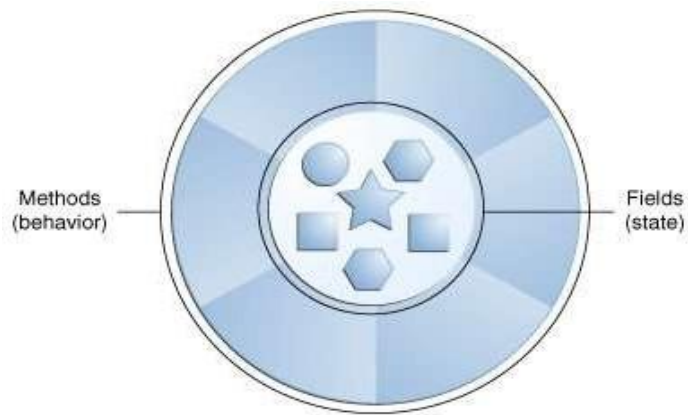
Why, you might wonder, use the intermediate Java bytecode at all? Why not just distribute the original Java program and let each person compile it into the machine language of whatever computer they want to run it on? There are many reasons. First of all, a compiler has to understand Java, a complex high-level language. The compiler is itself a complex program. A Java bytecode interpreter, on the other hand, is a fairly small, simple program. This makes it easy to write a bytecode interpreter for a new type of computer; once that is done, that computer can run any compiled Java program. It would be much harder to write a Java compiler for the same computer.

Features of JAVA

Object

Objects are key to understanding *object-oriented* technology. Look around right now and you'll find many examples of real-world objects: your dog, your desk, your television set, your bicycle.

Real-world objects share two characteristics: They all have *state* and *behavior*. Dogs have state (name, color, breed, hungry) and behavior (barking, fetching, wagging tail). Bicycles also have state (current gear, current pedal cadence, current speed) and behavior (changing gear, changing pedal cadence, applying brakes). Identifying the state and behavior for real-world objects is a great way to begin thinking in terms of object-oriented programming.



A software object.

Class

In the real world, you'll often find many individual objects all of the same kind. There may be thousands of other bicycles in existence, all of the same make and model. Each bicycle was built from the same set of blueprints and therefore contains the same components. In object-oriented terms, we say that your bicycle is an *instance* of the *class of objects* known as bicycles. A *class* is the blueprint from which individual objects are created.

The following `Bicycle` class is one possible implementation of a bicycle:

```
class Bicycle {
    int cadence = 0;
    int speed = 0;
    int gear = 1;

    void changeCadence(int newValue)
    { cadence = newValue;
    }

    void changeGear(int newValue)
    { gear = newValue;
    }

    void speedUp(int increment)
    { speed = speed +
    increment;
    }

    void applyBrakes(int decrement)
    { speed = speed - decrement;
    }
}
```

```

    }

    void printStates()
    { System.out.println("cadence:" +
    cadence + " speed:" +
    speed + " gear:" + gear);
    }
}

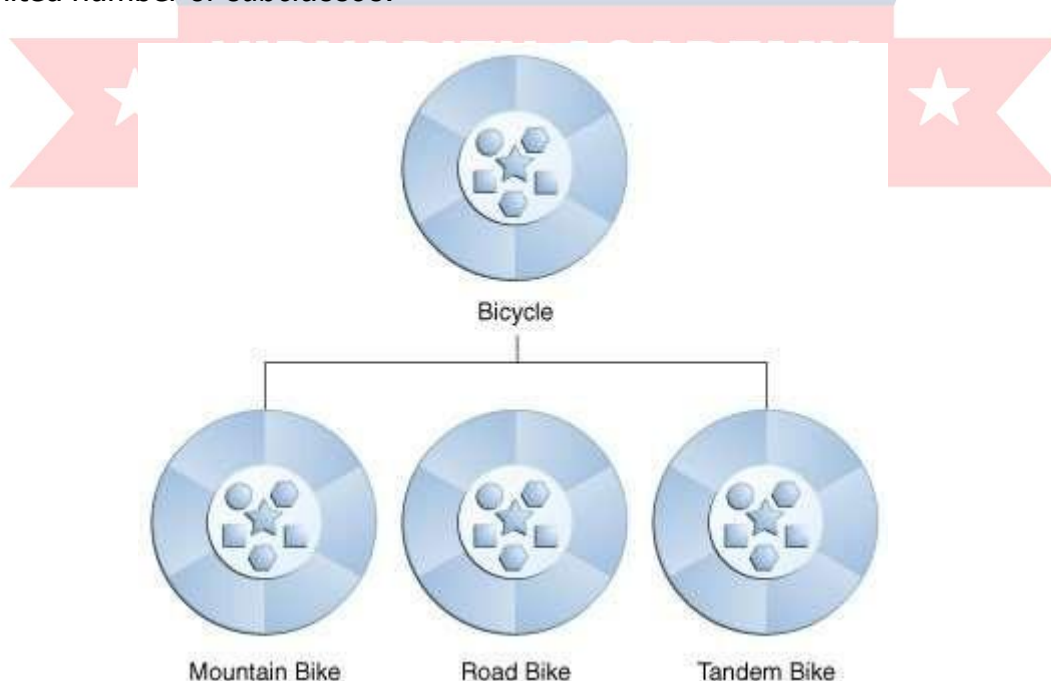
```

The syntax of the Java programming language will look new to you, but the design of this class is based on the previous discussion of bicycle objects. The fields `cadence`, `speed`, and `gear` represent the object's state, and the methods (`changeCadence`, `changeGear`, `speedUp` etc.) define its interaction with the outside world.

Inheritance

Different kinds of objects often have a certain amount in common with each other. Mountain bikes, road bikes, and tandem bikes, for example, all share the characteristics of bicycles (current speed, current pedal cadence, current gear). Yet each also defines additional features that make them different: tandem bicycles have two seats and two sets of handlebars; road bikes have drop handlebars; some mountain bikes have an additional chain ring, giving them a lower gear ratio.

Object-oriented programming allows classes to *inherit* commonly used state and behavior from other classes. In this example, `Bicycle` now becomes the *superclass* of `MountainBike`, `RoadBike`, and `TandemBike`. In the Java programming language, each class is allowed to have one direct superclass, and each superclass has the potential for an unlimited number of *subclasses*:



A hierarchy of bicycle classes.

The syntax for creating a subclass is simple. At the beginning of your class declaration, use the `extends` keyword, followed by the name of the class to inherit from:

```
class MountainBike extends Bicycle {  
  
    // new fields and methods defining  
    // a mountain bike would go here  
  
}
```

This gives `MountainBike` all the same fields and methods as `Bicycle`, yet allows its code to focus exclusively on the features that make it unique. This makes code for your subclasses easy to read. However, you must take care to properly document the state and behavior that each superclass defines, since that code will not appear in the source file of each subclass.

Interface

Methods form the object's *interface* with the outside world; the buttons on the front of your television set, for example, are the interface between you and the electrical wiring on the other side of its plastic casing. You press the "power" button to turn the television on and off.

In its most common form, an interface is a group of related methods with empty bodies. A bicycle's behavior, if specified as an interface, might appear as follows:

```
interface Bicycle {  
  
    // wheel revolutions per minute  
    void changeCadence(int newValue);  
  
    void changeGear(int newValue);  
  
    void speedUp(int increment);  
  
    void applyBrakes(int decrement);  
  
}
```

To implement this interface, the name of your class would change (to a particular brand of bicycle, for example, such as `ACMEBicycle`), and you'd use the `implements` keyword in the class declaration:

```
class ACMEBicycle implements Bicycle {  
  
    // remainder of this class
```

```
// implemented as before  
}
```

Implementing an interface allows a class to become more formal about the behavior it promises to provide. Interfaces form a contract between the class and the outside world, and this contract is enforced at build time by the compiler. If your class claims to implement an interface, all methods defined by that interface must appear in its source code before the class will successfully compile.

Package

A package is a namespace that organizes a set of related classes and interfaces. Conceptually you can think of packages as being similar to different folders on your computer. You might keep HTML pages in one folder, images in another, and scripts or applications in yet another. Because software written in the Java programming language can be composed of hundreds or *thousands* of individual classes, it makes sense to keep things organized by placing related classes and interfaces

Variables

As you learned in the previous lesson, an object stores its state in *fields*.

```
int cadence = 0;  
int speed = 0;  
int gear = 1;
```

The an Object discussion introduced you to fields, but you probably have still a few questions, such as: What are the rules and conventions for naming a field? Besides int, what other data types are there? Do fields have to be initialized when they are declared? Are fields assigned a default value if they are not explicitly initialized? We'll explore the answers to such questions in this lesson, but before we do, there are a few technical distinctions you must first become aware of. In the Java programming language, the terms "field" and "variable" are both used; this is a common source of confusion among new developers, since both often seem to refer to the same thing.

The Java programming language defines the following kinds of variables:

- **Instance Variables** (Non-Static Fields) Technically speaking, objects store their individual states in "non-static fields", that is, fields declared without the static keyword.
- **Class Variables** (Static Fields) A *class variable* is any field declared with the static modifier; this tells the compiler that there is exactly one copy of this variable in existence, regardless of how many times the class has been instantiated.
- **Local Variables** Similar to how an object stores its state in fields, a method will often store its temporary state in *local variables*.
- **Parameters** You've already seen examples of parameters, both in the Bicycle class and in the main method of the "Hello World!" application. Recall that the signature

for the main method is `public static void main(String[] args)`. Here, the `args` variable is the parameter to this method.

Data Types

The Java programming language is statically-typed, which means that all variables must first be declared before they can be used. This involves stating the variable's type and name, as you've already seen:

```
int gear = 1;
```

The eight primitive data types supported by the Java programming language are:

- **byte**: The byte data type is an 8-bit signed two's complement integer. It has a minimum value of -128 and a maximum value of 127 (inclusive).
- **short**: The short data type is a 16-bit signed two's complement integer. It has a minimum value of -32,768 and a maximum value of 32,767 (inclusive).
- **int**: The int data type is a 32-bit signed two's complement integer. It has a minimum value of -2,147,483,648 and a maximum value of 2,147,483,647 (inclusive).
- **long**: The long data type is a 64-bit signed two's complement integer. It has a minimum value of -9,223,372,036,854,775,808 and a maximum value of 9,223,372,036,854,775,807 (inclusive).
- **float**: The float data type is a single-precision 32-bit IEEE 754 floating point. Its range of values is beyond the scope of this discussion, but is specified in the Floating-Point Types, Formats, and Values section of the Java Language Specification.
- **double**: The double data type is a double-precision 64-bit IEEE 754 floating point.
- **boolean**: The boolean data type has only two possible values: `true` and `false`. Use this data type for simple flags that track true/false conditions. This data type represents one bit of information.
- **char**: The char data type is a single 16-bit Unicode character.

Literals

You may have noticed that the `new` keyword isn't used when initializing a variable of a primitive type. A *literal* is the source code representation of a fixed value; literals are represented directly in your code without requiring computation.

Integer Literals

An integer literal is of type `long` if it ends with the letter `L` or `l`; otherwise it is of type `int`. It is recommended that you use the upper case letter `L` because the lower case letter `l` is hard to distinguish from the digit `1`.

Integer literals can be expressed by these number systems:

- Decimal: Base 10

- Hexadecimal: Base 16
- Binary: Base 2

Floating-Point Literals

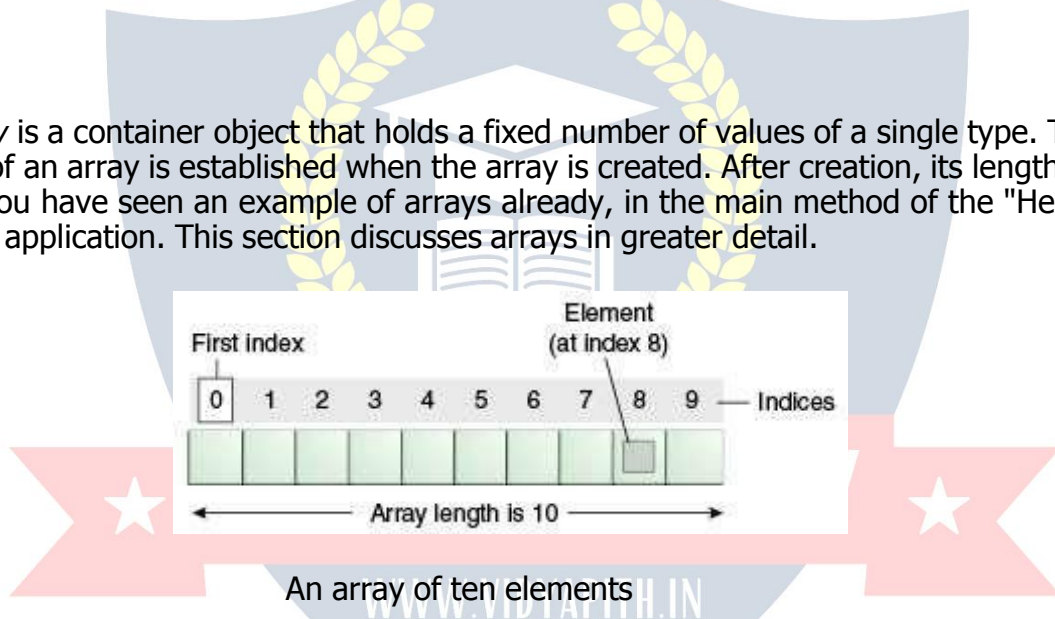
A floating-point literal is of type `float` if it ends with the letter `F` or `f`; otherwise its type is `double` and it can optionally end with the letter `D` or `d`.

Character and String Literals

Literals of types `char` and `String` may contain any Unicode characters. If your editor and file system allow it, you can use such characters directly in your code. If not, you can use a "Unicode escape". Always use 'single quotes' for `char` literals and "double quotes" for `String` literals. Unicode escape sequences may be used elsewhere in a program not just in `char` or `String` literals

Arrays

An *array* is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed. You have seen an example of arrays already, in the main method of the "Hello World!" application. This section discusses arrays in greater detail.



An array of ten elements

Each item in an array is called an *element*, and each element is accessed by its numerical *index*.

Declaring a Variable to Refer to an Array

The above program declares `anArray` with the following line of code:

```
// declares an array of integers int[]  
anArray;
```

Like declarations for variables of other types, an array declaration has two components: the array's type and the array's name.

Creating, Initializing, and Accessing an Array

One way to create an array is with the `new` operator. The next statement in the `ArrayDemo` program allocates an array with enough memory for ten integer elements and assigns the array to the `anArray` variable.

```
// create an array of integers anArray
= new int[10];
```

If this statement were missing, the compiler would print an error like the following, and compilation would fail:

`ArrayDemo.java:4: Variable anArray may not have been initialized.`
The next few lines assign values to each element of the array:

```
anArray[0] = 100; // initialize first element anArray[1]
= 200; // initialize second element anArray[2] = 300;
// etc.
```

Each array element is accessed by its numerical index:

```
System.out.println("Element 1 at index 0: " + anArray[0]);
System.out.println("Element 2 at index 1: " + anArray[1]);
System.out.println("Element 3 at index 2: " + anArray[2]);
```

Alternatively, you can use the shortcut syntax to create and initialize an array:

```
int[] anArray = {
100, 200, 300,
400, 500, 600,
700, 800, 900, 1000
};
```

Here the length of the array is determined by the number of values provided between `{` and `}`.

Operators

Operators are special symbols that perform specific operations on one, two, or three *operands*, and then return a result.

Operators	Precedence
postfix	<i>expr++ expr--</i>
unary	<i>++expr --expr +expr -expr ~ !</i>
multiplicative	<i>* / %</i>
additive	<i>+ -</i>
Shift	<i><< >> >>></i>
relational	<i>< > <= >= instanceof</i>
equality	<i>== !=</i>
bitwise AND	<i>&</i>
bitwise exclusive OR	<i>^</i>
bitwise inclusive OR	<i> </i>
logical AND	<i>&&</i>
logical OR	<i> </i>
ternary	<i>? :</i>
assignment	<i>= += -= *= /= %= &= ^= = <<= >>= >>>=</i>

Statements in JAVA

Control flow statements, however, break up the flow of execution by employing decision making, looping, and branching, enabling your program to *conditionally* execute particular blocks of code. This section describes the decisionmaking statements (**if-then**, **if-then-else**, **switch**), the looping statements (**for**, **while**, **do-while**), and the branching statements (**break**, **continue**, **return**) supported by the Java programming language.

The while and do-while Statements

The `while` statement continually executes a block of statements while a particular condition is `true`. Its syntax can be expressed as:

```
while (expression)
    {statement(s)}
}
```

The `while` statement evaluates *expression*, which must return a boolean value.

The Java programming language also provides a `do-while` statement, which can be expressed as follows:

```
do {
    statement(s) } while
(expression);
```

The difference between `do-while` and `while` is that `do-while` evaluates its expression at the bottom of the loop instead of the top.

The for Statement

The general form of the `for` statement can be expressed as follows:

```
for (initialization;      termination;
    increment) {
    statement(s)
}
```

When using this version of the `for` statement, keep in mind that:

- The *initialization* expression initializes the loop; it's executed once, as the loop begins.
- When the *termination* expression evaluates to false, the loop terminates.

- The *increment* expression is invoked after each iteration through the loop

ADVANCED JAVA

In the current era of networking, online transaction processing and managing the dataflow over network becomes an important issue. This subject is essential for providing knowledge and hands on experience over the issues of managing data on web, developing powerful GUI based friendly user interface, server side programming and developing applications for communication over network using object oriented fundamentals.

Advanced Java enhances the Java programming. After learning this subject, student will be able to develop network based software projects required in curriculum as well as industry.

To complete the Advance Java successfully, you should be familiar and have programming experience in basic Java programming. You must have good exposure to the object-oriented programming (OOP) concepts of Java language.

Objective of Advanced Java

- Create network based applications.
- Create business applications.
- Implement Server side programming.
- Develop dynamic software components.
- Develop database application.
- Design and develop powerful GUI based components.
- Create Animation using Applet, Thread and AWT controls.

The Internet, with particular respect to the World Wide Web, is growing at a tremendous rate. With over one million new pages going live every day, each one vying for our attention, greater emphasis is being placed on the servers delivering this information. The Internet is no longer a colorful brochure, where the user merely flicks from page to page. It has become a fully interactive experience, complete with inline video and stereo sound. The raise of server-side applications is one of the most exciting trends in Java programming. Java servlets are a key component of server-side Java development.

A servlet is a small, pluggable extension to a server that enhances the server's functionality. These servlets are commonly used with web servers, where they can take the place of CGI scripts. Servlets are to the server what applets are to the client. They extend the functionality of the web server in much the same way an applet extends the browser. To demonstrate the power of servlets, Sun developed the Java Web Server using the Java Server API. This is a complete web server, supporting both servlets and CGI, written entirely in Java. Implementing the web server in Java allows the web server to run on any platform with a Java Virtual Machine (JVM).

Common Gateway Interface (CGI)

The Common Gateway Interface popularly known as CGI, was one of the first techniques used to create dynamic content. A web server passes certain request to the external program with the help of CGI. The advent of CGI made it possible to create all sorts of new functionality in web pages. CGI defines a standard for communication between the web server and a separate program running on the same machine. CGI doesn't define anything else, and as a consequence, CGI scripts may be implemented in any language the platform will run. This lack of standardization at the server has meant CGI scripts implemented for one server platform have had to be redeveloped when required to run on another web server, on another platform.

Another problem area within CGI-based solutions is their overall efficiency. Every client request that requires processing by a CGI script spawns a separate program instance. This takes time. The operating system has to load the program, allocate memory for the program, and then deallocate and unload the program from memory. While the operating system is performing the housekeeping, nothing else can run. This is known as a heavyweight context switch. This is the reason CGI scripts are not suitable for applications that receive many client requests.

Java Server API

The foundation of the Server Toolkit is the Java Server API. This API allows the building of complete server-side applications. A servlet merely extends the functionality of a server, and using this API, the server may be built. The best example of a server built using this API is the Java Web Server from Sun. Although it is a very powerful and feature-rich API, it is not expected to be useful for everyone; instead, many developers will be more interested in the servlet API.

Java Servlet API

The servlet API allows the development of servlets. A servlet is designed to extend the functionality of the server it is running under. The servlet API comes as a set of classes that are used to form the base class for any user servlets. Any server supporting the servlet API will run any servlets developed using the servlet toolkit.

Servlet Overview

Java Servlets

Servlets are Java technology's answer to CGI programming. They are programs that run on a Web server and build Web pages. Building Web pages on the fly is useful (and commonly done) for a number of reasons:

- The Web page is based on data submitted by the user. For example the results pages from search engines are generated this way, and programs that process orders for e-commerce sites do this as well.
- The data changes frequently. For example, a weather-report or news headlines page might build the page dynamically, perhaps returning a previously built page if it is still up to date.

- The Web page uses information from corporate databases or other such sources. For example, you would use this for making a Web page at an on-line store that lists current prices and number of items in stock.

Basic Servlet Structure

Here's the outline of a basic servlet that handles GET requests. GET requests, for those unfamiliar with HTTP, are requests made by browsers when the user types in a URL on the address line, follows a link from a Web page, or makes an HTML form that does not specify a METHOD. Servlets can also very easily handle POST requests, which are generated when someone creates an HTML form that specifies METHOD="POST".

```
import java.io.*; import javax.servlet.*; import
javax.servlet.http.*; public class SomeServlet
extends HttpServlet { public void
doGet(HttpServletRequest request,
HttpServletResponse response) throws
ServletException, IOException {

// Use "request" to read incoming HTTP headers (e.g. cookies)
// and HTML form data (e.g. data the user entered and submitted)
// Use "response" to specify the HTTP response line and headers
// (e.g. specifying the content type, setting cookies).

PrintWriter out = response.getWriter();

// Use "out" to send content to browser

}

}
```

To be a Servlet, a class should extend `HttpServlet` and override `doGet` or `doPost` (or both), depending on whether the data is being sent by GET or by POST. These methods take two arguments: an `HttpServletRequest` and an `HttpServletResponse`. The `HttpServletRequest` has methods that let you find out about incoming information such as FORM data, HTTP request headers, and the like. The `HttpServletResponse` has

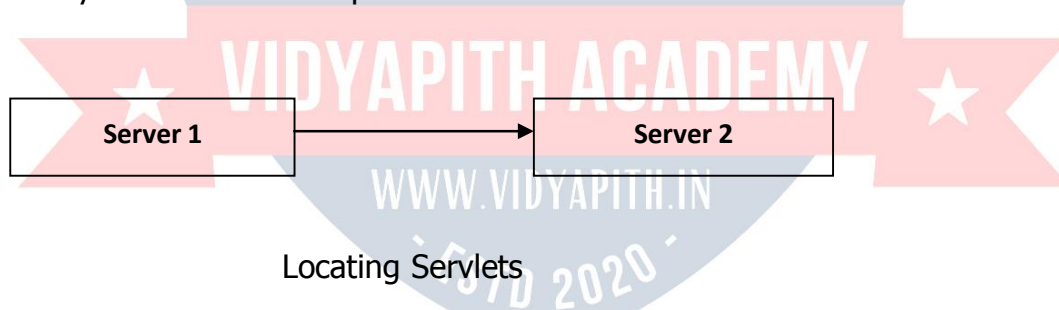
methods that lets you specify the HTTP response line (200, 404, etc.), response headers (Content-Type, Set-Cookie, etc.), and, most importantly, lets you obtain a `PrintWriter` used to send output back to the client. For simple Servlets, most of the effort is spent in `println` statements that generate the desired page. Note that `doGet` and `doPost` throw two exceptions, so you are required to include them in the declaration. Also note that you have to import classes in `java.io` (for `PrintWriter`, etc.), `javax.servlet` (for `HttpServlet`, etc.), and `javax.servlet.http` (for `HttpServletRequest` and `HttpServletResponse`). Finally, note that `doGet` and `doPost` are called by the service method, and sometimes you may want to override service directly, e.g. for a Servlet that handles both GET and POST request.

The Life Cycle of a Servlet

Since servlets come in the form of Java objects, there are many different variations on not only how they are loaded but also how they are unloaded again, if at all. When the server decides a particular servlet is to be loaded, it uses the standard Java class loading mechanisms to create the class instance. Using this technique, servlets can be loaded from anywhere on the network, and if the server is connected to the Internet, from anywhere in the world.

```
Class c = Class.forName( "http://<server>/testclass");
```

Once this method has returned, the class can be accessed as normal. Assuming the URL doesn't upset the security of the system, classes can be located anywhere on the network. Forgetting about the potential security implications this may have provides a great advantage to distribution and resource sharing. Unlike the applet-loading model, where every client running the applet loads the applet from the same origin, the servlet has the ability to load from multiple hosts.



The actual loading and execution of the servlet is straightforward enough--when a client connection is accepted and the servlet is not in memory, it is loaded. However, a servlet can be loaded if it has not yet been addressed. A servlet can be loaded either at server startup or dynamically when it is accessed. The Java Web Server has an administration section that allows the administrator to specify which servlets are loaded at startup. This allows the servlet to be ready in memory for the first client request to come in. Generally, only servlets that are expected to be heavily use dare loaded at server startup. Loading the servlet at startup ensures the response time for all requests is kept to a minimum, as opposed to waiting for the servlet to be loaded.

A Simple Servlet Generating Plain Text

Here is a simple Servlet that just generates plain text. The following section will show

the more usual case where HTML is generated. HelloWorld.java package hall;

```
import java.io.*; import javax.servlet.*; import javax.servlet.http.*; public class
```

```
HelloWorld extends HttpServlet { public void doGet(HttpServletRequest request,
```

```
HttpServletResponse response) throws ServletException, IOException {
```

```
PrintWriter out = response.getWriter(); out.println("Hello World");
```

```
}
```

```
}
```

Compiling and installing the Servlet

Servlet is placed in a separate package (hall) to avoid conflicts with other Servlets on this server; you'll want to do the same if you are using a Web server that is used by other people and doesn't have a good infrastructure for "virtual servers" to prevent these conflicts automatically. Thus, Hello World.java actually goes in a subdirectory called hall in the Servlets directory. Note that setup on most other servers is similar, material.

Setting Class.Java

One way is to set your CLASSPATH to point to the directory above the one actually containing your Servlets. You can then compile normally from within the directory. For example, if your base directory is C:\JavaWebServer\servlets and your package name (and thus subdirectory name) is hall, and you were on Windows, you'd do:

```
DOS> set CLASSPATH=C:\JavaWebServer\servlets;%CLASSPATH% DOS>
```

```
cd C:\JavaWebServer\servlets\hall DOS> javac YourServlet.java
```

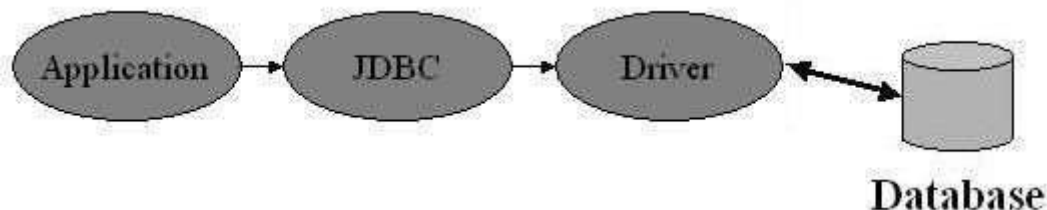
The first part, setting the CLASSPATH, you probably want to do permanently, rather than each time you start a new DOS window. On Windows 95/98 you'd typically put the "set CLASSPATH=..." statement in your autoexec.bat file somewhere after the line that set the CLASSPATH to point to servlet.jar and jsp.jar. On Windows NT, you'd go to the Start menu, select Settings, select Control Panel, select System, select Environment, then enter the variable and value. Note also that if your package were of the form name1.name2.name3 rather than simply name1 as here, you'd still have the CLASSPATH point to the top-level directory of your package hierarchy (the one containing name1). A second way to compile classes that are in packages is to go to the directory above the one containing your Servlets, and then do "javac

directory\YourServlet.java" (Windows; note the backslash) or "javac directory/YourServlet.java" (Unix; note the forward slash).

Java JDBC

The JDBC (Java Database Connectivity) API defines interfaces and classes for writing database applications in Java by making database connections. Using JDBC you can send SQL, PL/SQL statements to almost any relational database. JDBC is a Java API for executing SQL statements and supports basic SQL functionality. It provides RDBMS access by allowing you to embed SQL inside Java code. Because Java can run on a thin client, applets embedded in Web pages can contain downloadable JDBC code to enable remote database access. You will learn how to create a table, insert values into it, query the table, retrieve results, and update the table with the help of a JDBC Program example. Although JDBC was designed specifically to provide a Java interface to relational databases, you may find that you need to write Java code to access nonrelational databases as well.

JDBC Architecture



Java application calls the JDBC library. JDBC loads a driver which talks to the database. We can change database engines without changing database code.

JDBC Basics – Java Database Connectivity Steps

Before you can create a java jdbc connection to the database, you must first import the java.sql package. `import java.sql.*;`

The star (*) indicates that all of the classes in the package java.sql are to be imported.

Loading a database driver,

In this step of the jdbc connection process, we load the driver class by calling `Class.forName()` with the Driver class name as an argument. Once loaded, the Driver class creates an instance of itself. A client can connect to Database Server through JDBC Driver. Since most of the Database servers support ODBC driver therefore JDBC-ODBC Bridge driver is commonly used. The return type of the `Class.forName (String ClassName)` method is "Class". Class is a class in `java.lang` package.

```
try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); //Or any other driver
}
catch(Exception x){
    System.out.println("←nable to load the driver class!");
}
```

Creating a oracle jdbc Connection

The JDBC DriverManager class defines objects which can connect Java applications to a JDBC driver. DriverManager is considered the backbone of JDBC architecture. DriverManager class manages the JDBC drivers that are installed on the system. Its `getConnection()` method is used to establish a connection to a database. It uses a username, password, and a jdbc url to establish a connection to the database and returns a connection object. A jdbc Connection represents a session/connection with a specific database. Within the context of a Connection, SQL, PL/SQL statements are executed and results are returned. An application can have one or more connections with a single database, or it can have many connections with different databases. A Connection object provides metadata i.e. information about the database, tables, and fields. It also contains methods to deal with transactions.

Creating a jdbc Statement object,

Once a connection is obtained we can interact with the database. Connection interface defines methods for interacting with the database via the established connection. To execute SQL statements, you need to instantiate a Statement object from your connection object by using the `createStatement()` method.

```
Statement statement = dbConnection.createStatement();
```

A statement object is used to send and execute SQL statements to a database.

Three kinds of Statements

Statement: Execute simple sql queries without parameters.

Statement createStatement()

Creates an SQL Statement object.

Prepared Statement: Execute precompiled sql queries with or without parameters.

PreparedStatement prepareStatement(String sql) returns a new PreparedStatement object.

PreparedStatement objects are precompiled SQL statements.

Callable Statement: Execute a call to a database stored procedure. CallableStatement

prepareCall(String sql)

returns a new CallableStatement object. CallableStatement objects are SQL stored procedure call statements.

Executing a SQL statement with the Statement object, and returning a jdbc resultSet.

Statement interface defines methods that are used to interact with database via the execution of SQL statements. The Statement class has three methods for executing statements:

executeQuery(), executeUpdate(), and execute(). For a SELECT statement, the method to use is executeQuery . For statements that create or modify tables, the method to use is executeUpdate. Note: Statements that create a table, alter a table, or drop a table are all examples of DDL

statements and are executed with the method executeUpdate. execute() executes an SQL statement that is written as String object.

ResultSet provides access to a table of data generated by executing a Statement. The table rows are retrieved in sequence. A ResultSet maintains a cursor pointing to its current row of data. The next() method is used to successively step through the rows of the tabular results.

ResultSetMetaData Interface holds information on the types and properties of the columns in a ResultSet. It is constructed from the Connection object.

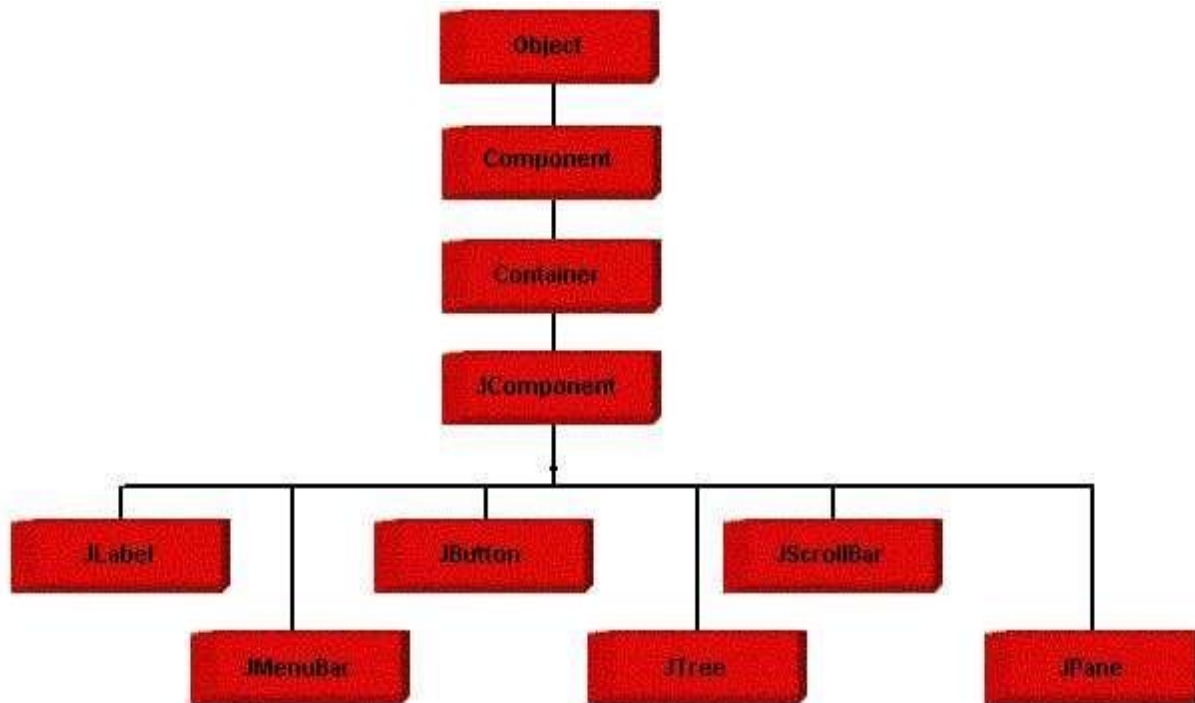
Swings in Java

Swing, which is an extension library to the AWT, includes new and improved components that enhance the look and functionality of GUIs. Swing can be used to build Standalone swing gui Apps as well as Servlets and Applets. It employs a model/view design architecture. Swing is more portable and more flexible than AWT.

Swing Model/view design: The "view part" of the M ↑ design is implemented with a component object and the ←I object. The "model part" of the M ↑ design is implemented by a model object and a change listener object.

Java Swing class hierarchy

The class **JComponent**, descended directly from Container, is the root class for most of Swing's user interface components. Swing contains **components** that you'll use to build a GUI. I am listing you some of the commonly used Swing components. To learn and understand these swing programs, AWT Programming knowledge **is not** required.



Heirarchy of JComponent

Swing is built on top of AWT and is entirely written in Java, using AWT's lightweight component support. In particular, unlike AWT, the architecture of Swing components makes it easy to customize both their appearance and behavior. Components from AWT and Swing can be mixed, allowing you to add Swing support to existing AWT-based programs. For example, swing components such as JSlider, JButton and JCheckbox could be used in the same program with standard AWT labels, textfields and scrollbars. You could subclass the existing Swing UI, model, or change listener classes without having to reinvent the entire implementation. Swing also has the ability to replace these objects on-the-fly. In Swing, classes that represent GUI components have names beginning with the letter J. Some examples are JButton, JLabel, and JSlider. Altogether there are more than 250 new classes and 75 interfaces in Swing — twice as many as in AWT.