Basic Computer Engineering
BT 205
Lab Manual

Index

Beyond Syllabus:

Aim 1

Aim:- Study and practice of Internal & External DOS Commands.

Theory:- DOS Commands are instructions to perform tasks on files and directories very useful to Windows users. DOS commands are case insensitive. File is the area where we store group of information or data, and collection of group of files is called a directory.

A.Internal commands: The internal commands are those commands that are automatically loaded in the memory. Some commonly used DOS internal commands are

1 Cls
2. Dir
3. Date
4. Time
5. Ver
6. Copycon
7. Type
8. Ren
9. Del
10. MD
11. CD
12. RD
13. Copy

1) Cls :- The purpose of this command is to clear the display screen and redisplay the Dos prompt at the top left corner of the screen.

Syntax:- C : / > Cls

2) Dir:- It displays the list of directories and files on the screen.

Syntax:- C : / > dir.

a. C : / > dir/p – It displays the list of directories or files page wise
b. C: / > dir/w- It displays the list of directories or files width wise
c. C : / > dir/d: –It display list of directories or files in drive D
d. C : / > dir filename **.** extension – It displays the information of specified file.
e. C : / > dir file name with wild cards.

Wild cards: - It is the set of special characters wild are used with some commonly used DOS commands there are two types of wild cards.

1. Asterisk (*˜)
2. Question mark (?)

1. Asterisk:- (*) The wild word will match all characters.
1. C : / > dir *.* - will display list of all files and directories.
2. C : / > dir R*.* - will display all files stored with first character R.

2. Question mark: - This wild card represents a single character that a group or files have in common.

1) C : / > dir ac .* ˜- will display all files having any first character and remaining name has given in command.

2) C : / > dir ??? R **.** doc-will display all files having extension doc and having any first three letters and fourth letter is R.

3) Date: - It displays the current system date. User can also change the current date with new date by using this command.

Syntax: - C : /> date
Current date is: sat 3-25-2015
Enter of new date (mm-dd-yy):-

4. Time : - It displays the current systems time user can also change existing time with new time by using this command.

Syntax : - C : / > time
Current time is 12 : 39 - 48 : 36 p
Enter new time : -

5. VER : It displays the version of DOS being used currently.

Syntax : - C : / > Ver
MS – Dos version 6 : 20
Copy card .

6. copycon : - The purpose of this command is to create a file.

Syntax : - C : / > copy con filename **.** extension

Saves the contents of file by pressing ctrl +z key combination at the last time of the file. File name should not be greater than 11 characters out of which 8 characters are for the name and 3 characters are for the extension.

Extension is optional :
Eg : C : / > copy con ram
I am a good boy
1. File is copied.
C : / >

7. TYPE:- Allows the user to see the contents of a file.

Syntax :- C :/ >Type path
Eg: C:/ > Type D:/> ramu

8. REN : - The purpose of this command is to rename the old file name with new file name.

Syntax : - C : / > ren oldfilename newfilename
C : / > ren ramu somu

9. DEL:- The purpose of this command is to delete file. The user can also delete multiple files by busing this command and long with while cards.

Syntax : - C : / > Del file name **.** extension
C : / > Del ramu
C : Del x **.** prg.

10. MD:- The purpose of this command is to create a new directly or sub directly i.e sub ordinate to the currently logged directly.

Syntax : - C : /> MD directory
C : /> MD sub directory

Ex : C : / > MD college
Now user wants to create a sub directory first year in college directory then
C : / > cd college
C : / > college > Md first year

11. CD : - The purpose of this command is to change from one director to another directory or sub – directory.

Syntax : - C : / > CD directory name

Ex : C: / > cd college
C : / > college > CD first year
C : / > college > first year >
If the user wants to move to the parents directory then use CD command as
C : / > college > first year > cd ….
C : / > college >

12 . RD: - The purpose of this command is to remove a directory or sub directory. If the user wants to remove a directory or sub – directory then first delete all the files in the sub – directory and then remove sub directory and remove empty main directory.

13. COPY: The purpose of this command is to copy one or more specified files to another disk with same file name or with different file name.

Syntax : - C : / > copy source path target path
C : / > copy A :/> ˜ *. * ˜ C : /> chinni

<u>B. External commands</u>: - This commands are not permanent part of the memory. To execute or run this commands an external file is required.

Example : [.] Dot exe, bat.

Some commonly used DOS external commands are .

1. CHKDSK:- The command CHSDK returns the configuration status of the selected disk. It returns the information about the volume, serial number, total disk space, space in directories, space in each allocation unit, total memory and free memory.

Syntax : - C : / > CHKDSK drive name
Eg:- C : / > CHKDSK e :
If drive name is not mentioned by default current drive is considered.

2. Diskcopy : - Disk copy command is used to make duplicate copy of the disk like Xerox copy. It first formats the target disk and then copies the files by collection. From the source disk and copied to the target disk.

Syntax : - C : / > disk copy < source path > < destination path >
Ex: - c : / > diskcopy A : B :

NOTE: - This command is used after diskcopy command to ensure that disk is copied successfully.

3. Format : - Format is used to erase information off of a computer diskette or fixed drive.

Syntax : - C : / > format drive name
Ex : C : / > format A:

4. Label : This command is used to see volume label and to change volume label.

Syntax : C : / > label drive name
Ex : C : / > label A:

5.Scandisk : - This utility is used to repair and check various disk errors. It also defects various physical disk errors and surface errors.
Syntax : **-** C : / > scandisk < drive names >
 C : / > Scandisk A :

6. Move : The purpose of move is move to files from one place to another place.

Syntax: C : / > Move < source path > < target path >

7. Print : This command allowed users to print a text file to a line printer.
Syntax : C : / > Print < files name >
C : / > print ramu

8. Tree : This command displays the list of directories and files on specified path using graphical display. It displays directories of files like a tree.

Syntax:- C : / > tree > path
C : / > tree A:

9. Deltree: This command is used to delete files a directories same as by the del and RD commands. This command is more useful than del and RD commands because it completely removes specified directories ie., disk will all it files and sub – directories at a time.

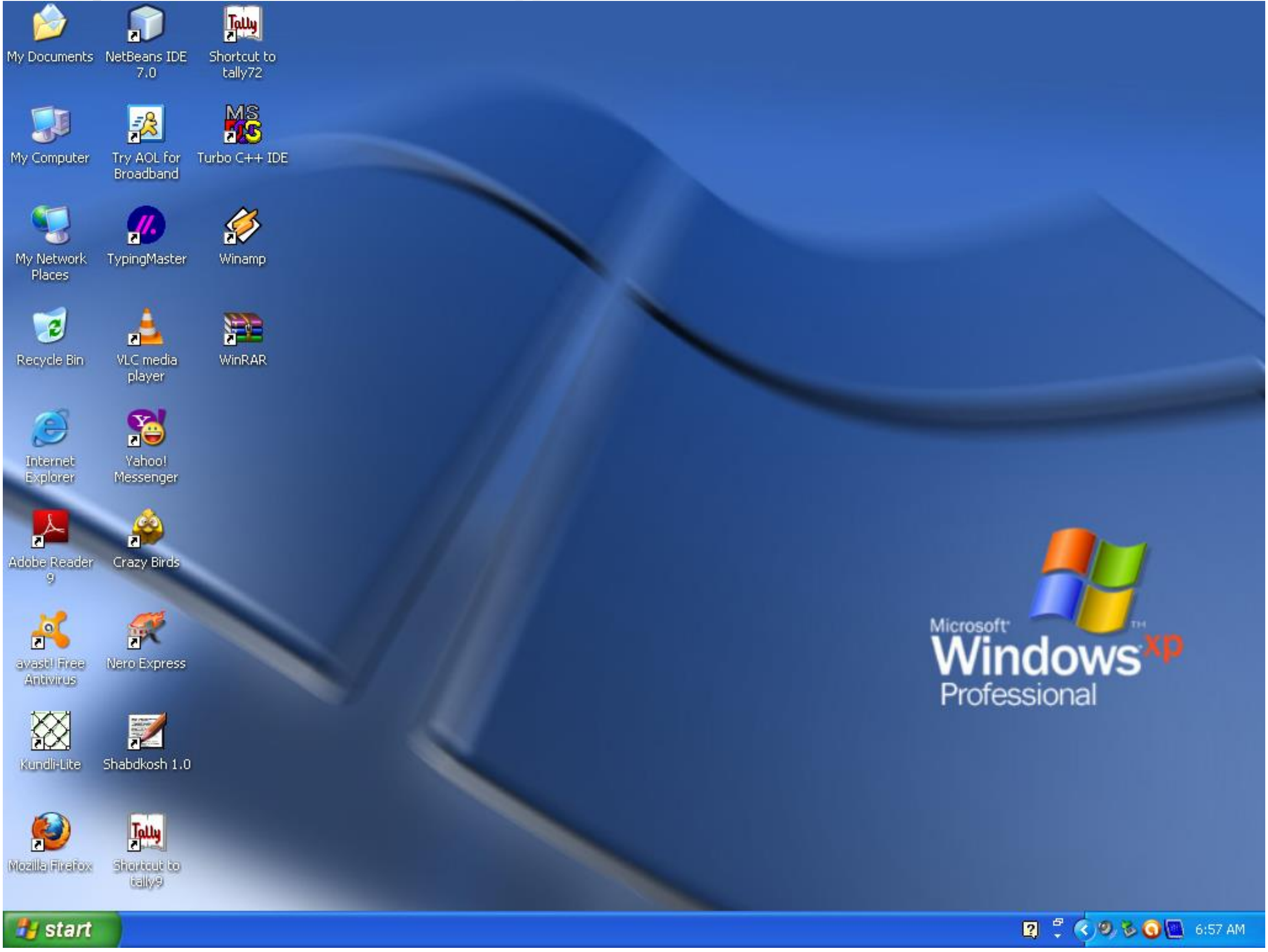
Syntax:- C : / > deltree (path)

C : / > deltree A:/>ramu

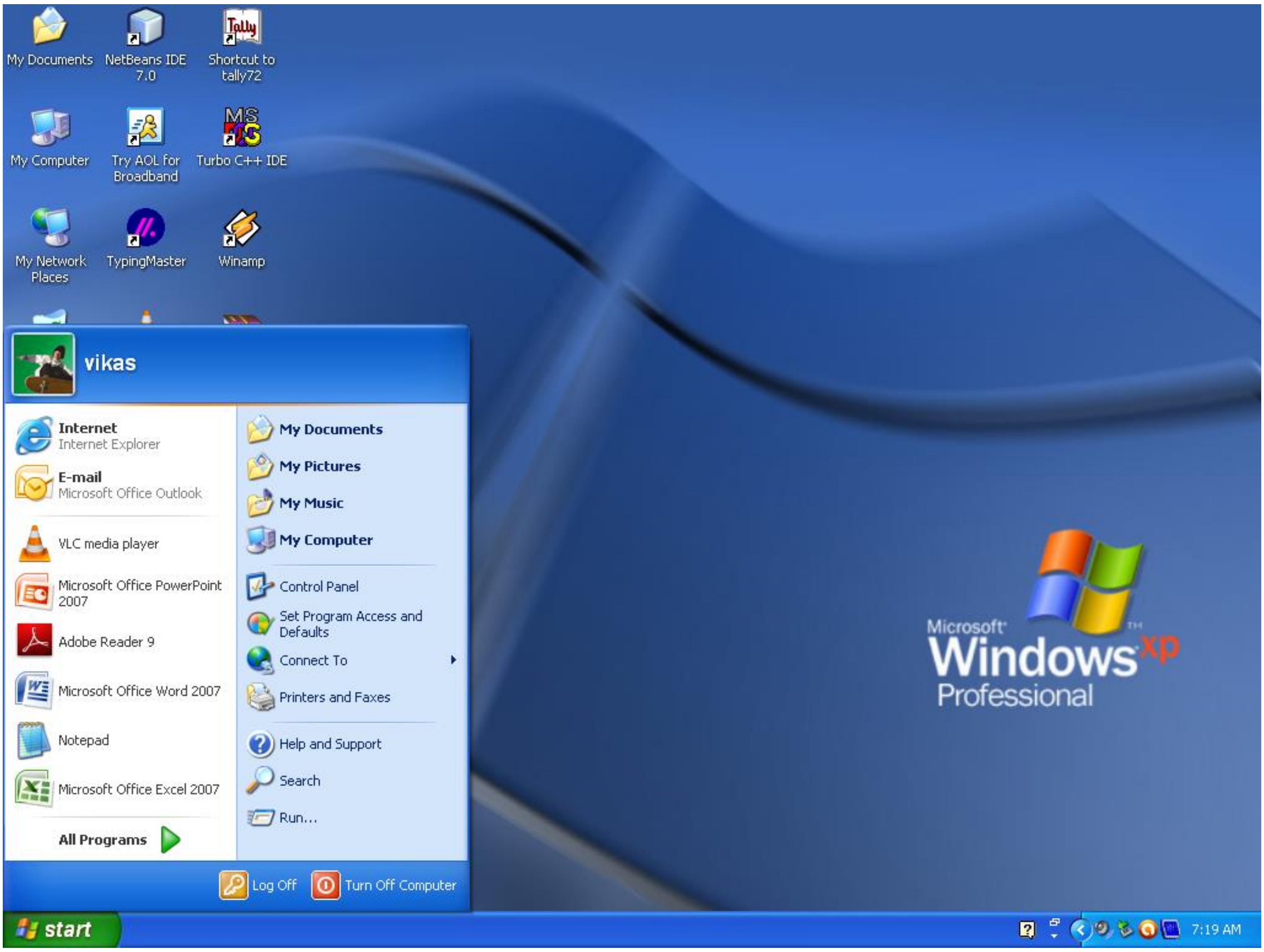
Aim 2

Aim:-Study and practice of MS windows- Folder related operations, My-Computer,window explorer, Control Panel.

Desktop:- The small picture on the desktop or any other window are called Icon. Icon are the gateway to program, application, files and folders started on our computer. We can open any program or files by double clicking on an icon. Icon need not only be on desktop. İcon will be seen in all the applicaion and window.



Start Menu:-Window Explorer is a file manager application that is included with releases by the Microsoft window operating system. It provides a Graphical User Interface (GUI) for accessing the file system. It is also the component of the operating system that present many user interface items on the monitor such as the taskbar and desktop. Start menu provides all shortcut link of the desktop icons such as My computer, My desktop, My music, My picture, Contro; panel, search,run and many more which is recently used by user.



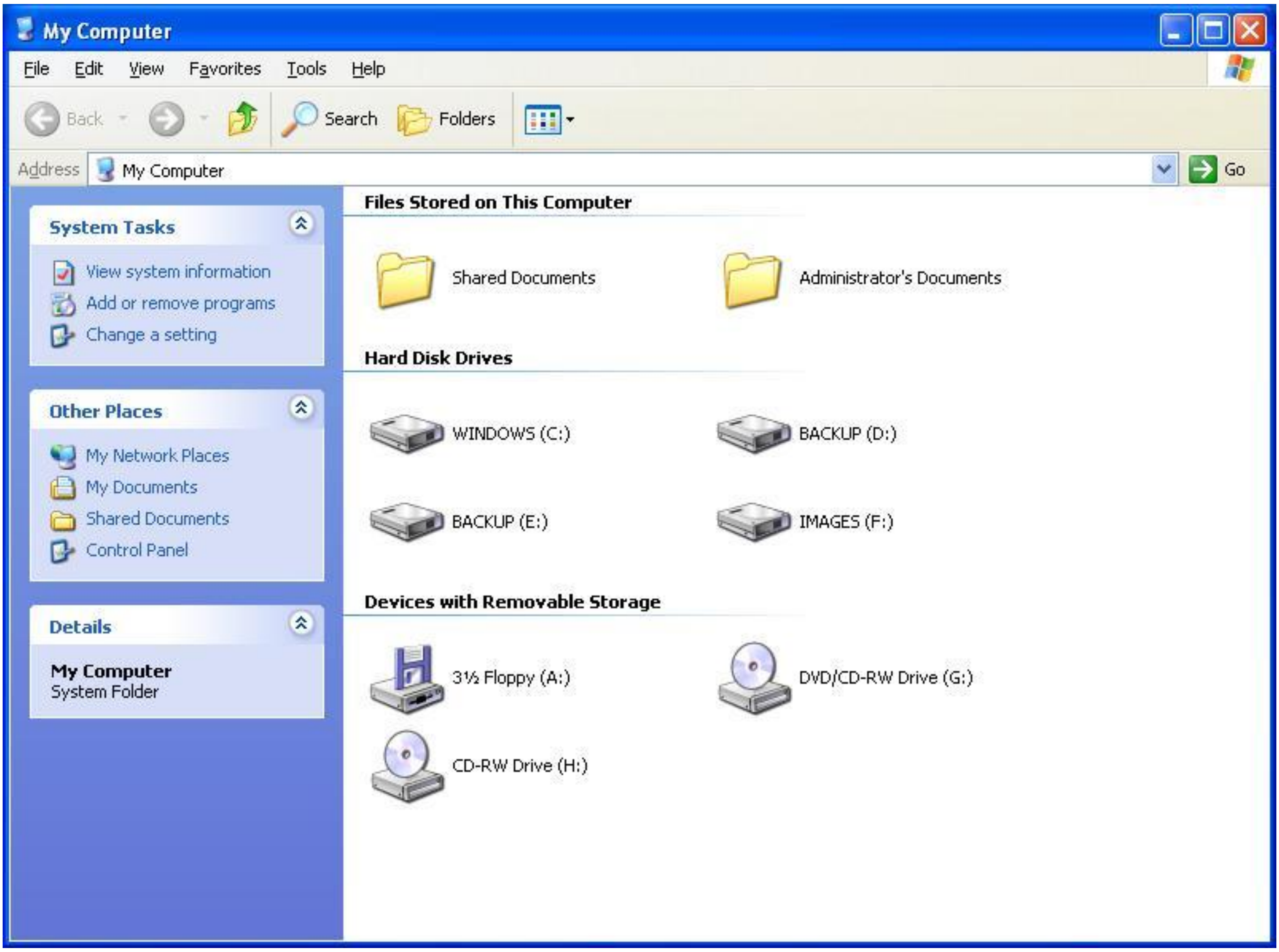
Folder option:-This option is obtained in window in menu bar in that tools option. The Folder option could be seen there. This specifies how folders are displayed. There is provision to view hidden files also. We could easily see the extension of any kind of file.

My Computer:-My computer window displays all the contents of your computer disk drives on two panels. We can open this wndow by double clicking the my computer icon on desktop or select my computer from start menu option.

The right panel of this window displays the content of your floppy disk, hard disk, CD-ROM drive and network drive attached to the system. Infact, now-a-days we can attached removable drives like the pen drive.

The left panel give us four important links:-

1). My Network Place
2). My Document
3). Shared Documents
4). Control Panel



The documents, data and files which use save by default they are saved in my document folder which can be easily access. "My Document" is created by windows itself in the c: drive. Using these above links we can search for and open files and folders on our computer.

Menu Bar:-Commonly there are six contents in Menu Bar of My Computer:-

1). **File:-**The first menu of My Computer is 'File'. It allows to delete, rename files. It can create shortcuts to a particular folder and to view the properties of a drive. It allows to close a file.

2). **Edit:-** It allows the user to cut,copy,paste,move a folder or a file. Undo allows to back a done process. It also allows select all contents in the file.

3). **View:-**It is used to arrange the icons or file. It contains tool bar, status bar, explorer bar, thumbnails, arrange icon, choose details, go to several drives and to refresh.

4). **Favorites:-**To create a special criteria of some special files and the most used files. İt contains Add to favorites, organize favorites links.

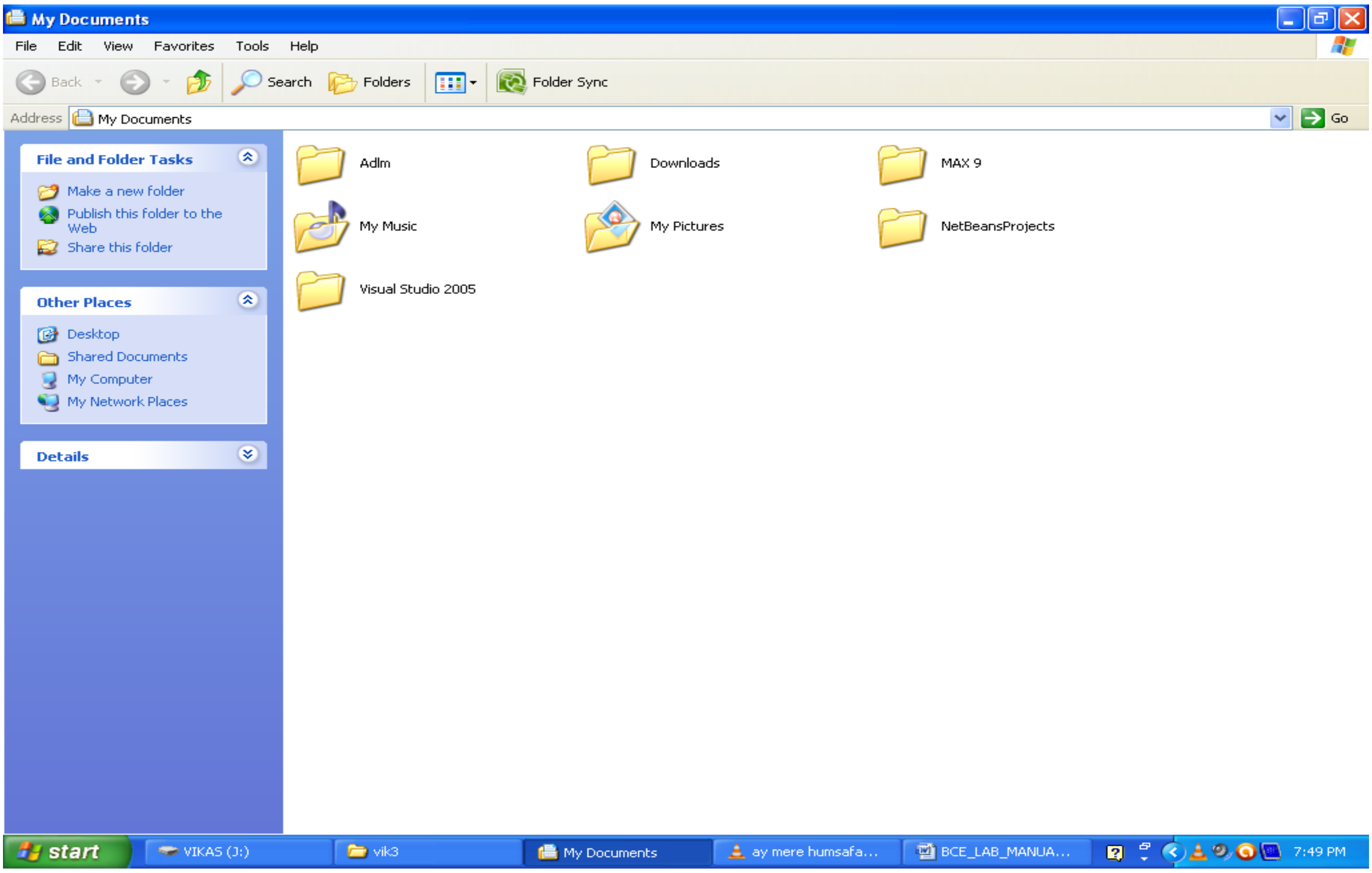
5). **Tools:-** It contains the settings, configuration which is used in folders an it has many other file folder options.

6). **Help:-** it is used to understand some difficult process. İt contains knowledge about the Operating system and of folders. It provide help to use to know the process.

Adjust system settings with your computer:-

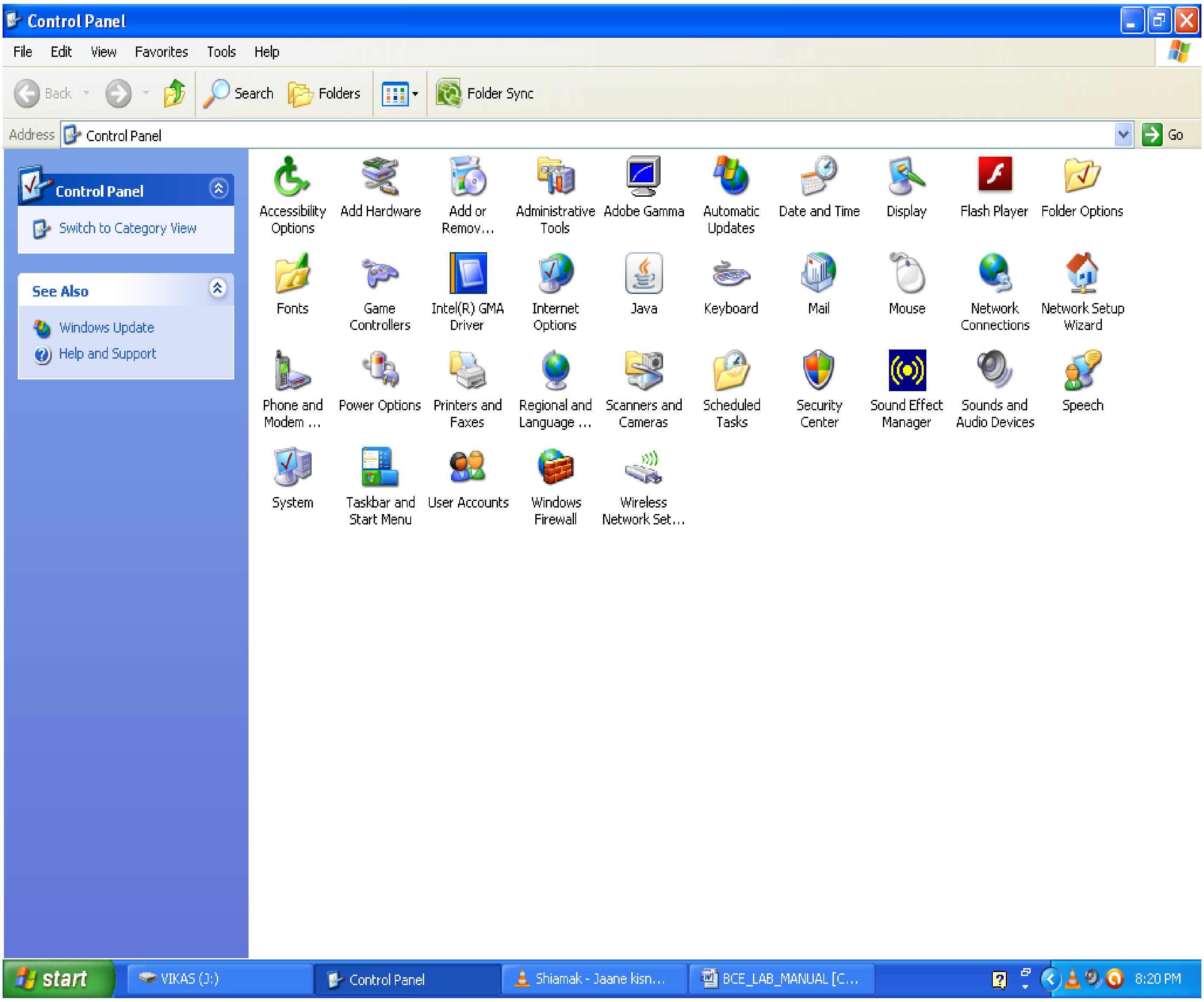
If you wish to manage your computer or view other settings and information about your computer instead of double-clicking the My Computer icon to open it, right-click on the My Computer icon and click Properties. Performing these steps will open your System Properties.

My Document:-My document is a folder within the windows operating system that provides us with a convenient place to store documents, graphics or other files we want to access quickly. When we save a file in a program such as Microsoft- word, Microsoft excel, WordPad or paint, the file is automatically saved in My Document, unless we choose a different folder.



Control panel:-The control panel is a feature of the window operating system that allows the user to modify system setting and control. It includes several applications or control panel that can be used to view and change hardware or software settings. Some example of hardwarecontrol panel are display, keyboard and mouse settings. Software control panel includes date and time, power option, fonts and administraor tools.

File:-A file is a collection of data stored in one unit, identified by a file name. It can be a Document, picture, audio or video stream, data library, application or other collection of data. Files can be opened, saved, deleted and moved to different folders. They can also be transferred across network connections or downloaded from the internet. A file type can be determined by viewing the file icon or by reading the file extension. İf the file type is associated with a specific application, double clicking the file file will typically open the file within the program.



<u>Drives</u>:-Computer drives are an integral part of any computer system, be it a PC or a Mac. They perform a variety of different functions ranging from storage to CD and DVD input, through input and output of different types of data, they give us the flexibility to customize and control the funtionality of our computer systems.

Computer drives caome in many sizes and shapes. They allow the user to input specific types of information to be stored in the computers memory. Many drives are input only, meaning they can be used only for reading the information. Other drives such as CD and DVD writers can output information to physical media.

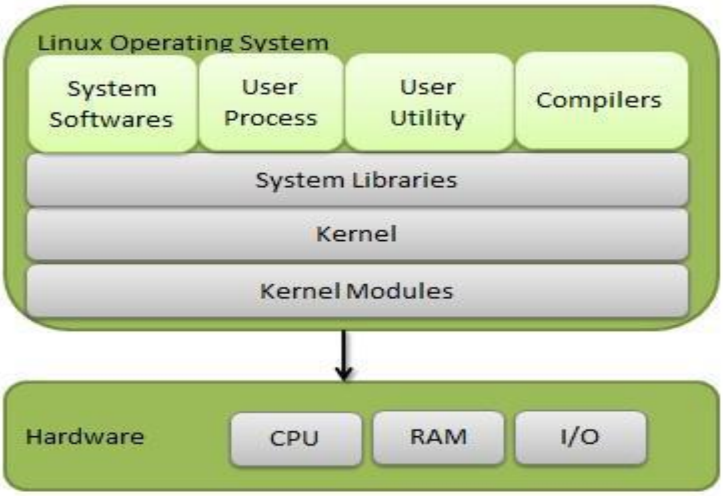
Aim 3

Aim:- Study and practice of Basic Linux Commands- ls,cp,mv,chmod,kill,ps etc.

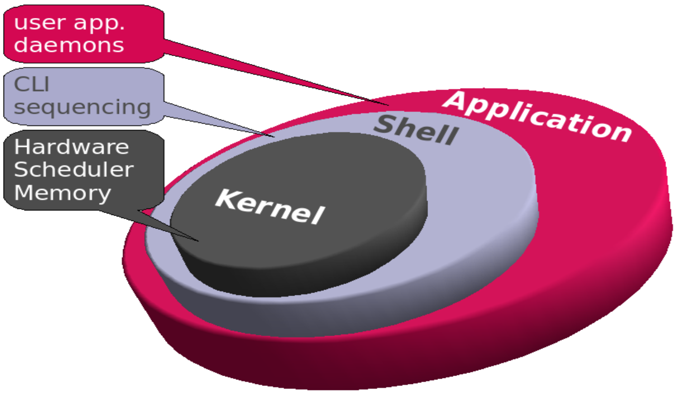
Theory:- Linux is a MULTIUSER, MULTIPROGRAMMING Operating System, primarily consist of three components:-

Kernel - Kernel is the core, interacts directly with the underlying hardware.
System Library - System libraries are programs using which application programs or system utilities accesses Kernel's features.

System Utility - System Utility programs are responsible to do specialized, individual level tasks.



Following are some of the important features of Linux Operating System.
Portable - Linux kernel and application programs supports their installation on any kind of hardware platform.
Open Source - Linux source code is freely available
Multi-User - Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time**.**
Multiprogramming - Linux is a multiprogramming system means multiple applications can run at same time.



Linux is available in several distributions.
1. Red Hat Enterprise.
2. CentOS.
3. Debian.
4. OpenSUSE.
5. Ubuntu.
Linux also supports several hardware platforms:
1. Intel,
2. PowerPC,
3. DEC Alpha,
4. Sun Sparc, and
5. Motorola.

Basic Linux Commands are:

1. ls : Display filesize in human readable format (e.g. KB, MB etc.,)

```
$ ls –lh
```

2. ps : display information about the processes that are running in the system.

```
$ ps -ef | more
$ ps -efH | more
```

3. mkdir : make directory - command allows the user to make a new directory.

```
$mkdir ggits
```

4. cat > filename : cerate new file

```
$cat > cse1
```

5. cp : Copy file1 to file2

```
$ cp -p file1 file2
```

6. mv : Rename file1 to file2.

```
$ mv -i file1 file2
```

7. rmdir: remove directory - command allows the user to remove an existing directory

e.g- rmdir ggits

8. rm : Deletes the file in the current directory.

9. date : displays the current system date.

10. clear : clear command clears the screen and wipes the board clean

11. chmod - To change access permissions, change mode.

```
$chmod [Options]... Numeric_Mode file...
```

The format of a numberic mode:

A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Any omitted digits are assumed to be leading zeros. The first digit selects the set user ID (4) and set group ID (2) and sticky (1) attributes. The second digit selects permissions for the user who owns the file: read (4), write (2), and execute (1); the third selects permissions for other users in the file's group, with the same values; and the fourth for other users not in the file's group, with the same values.

Read by owner only

```
$ chmod 400 sample.txt
```

Read by group only

```
$ chmod 040 sample.txt
```

Read by anyone

```
$ chmod 004 sample.txt
```

Write by owner only

```
$ chmod 200 sample.txt
```

Write by group only

```
$ chmod 020 sample.txt
```

Write by anyone

```
$ chmod 002 sample.txt
```

Execute by owner only

```
$ chmod 100 sample.txt
```

Execute by group only

```
$ chmod 010 sample.txt
```

Execute by anyone

```
$ chmod 001 sample.txt
```

Allow read permission to owner and group and anyone.

```
$ chmod 444 sample.txt
```

Allow everyone to read, write, and execute file.

```
$ chmod 777 sample.txt
```

Symbolic mode

Multiple symbolic operations can be given, separated by commas. A combination of the letters 'ugoa' controls which users' access to the file will be changed: the user who owns it (u), other users in the file's group (g), other users not in the file's group (o), or all users (a). If none of these are given, the effect is as if 'a' were given, but bits that are set in the umask are not affected. The operator '+' causes the permissions selected to be added to the existing permissions of each file; '-' causes them to be removed; and '=' causes them to be the only permissions that the file has.

The letters 'rwxXstugo' select the new permissions for the affected users: read (r), write (w), execute (or access for directories) (x), execute only if the file is a directory or already has execute permission for some user (X), set user or group ID on execution (s), sticky (t), the permissions granted to the user who owns the file (u), the permissions granted to other users who are members of the file's group (g), and the permissions granted to users that are in neither of the two preceding categories (o).

EXAMPLES

Deny execute permission to everyone.

```
$ chmod a-x sample.txt
```

Allow read permission to everyone.

```
$ chmod a+r sample.txt
```

Make a file readable and writable by the group and others.

```
$ chmod go+rw sample.txt
```

Make a shell script executable by the user/owner.

```
$ chmod u+x samplescript.sh
```

Allow everyone to read, write, and execute the file and turn on the set group-ID.

```
$ chmod =rwx,g+s samplescript.sh
```

12. kill : The kill command is usually used to kill a process. Internally it sends a signal, and depending on what you want to do, there are different signals that you can send using this tool.

```
$ kill [options] <pid> [...]
```

Send a Kill Process ID 1414

```
$ kill -KILL 1414
```

To stop all of your processes and log yourself off, enter the following command

```
$ kill -kill
```

Aim 4

Aim:- Cration and editing of text files using MS-Word.

Theory:- Introduction of MS Word

In this section. We will discus various tools, function and commands of MSword 2007 software to create an open, same, front and close, save, print and close the word document Before staring work on it, the Ms office needs to be installed on the computer system. After installing the Ms office. The following steps are required to access the Ms word.

We know that there are two kinds of software : System software and application s/w Ms office is a very useful application software package consisting of several application program such as Ms word, Ms excel, Ms power point, Ms Access etc out of these, ms word is the most popular word processing software is used to create, edit, save and print a document on the windows OS environment it provides a Graphical User Interface (GUI) which makes word processing very simple and efficient through availability of various commands.

Before starting work on it, the Ms office needs to be installed on the computer system. After installing the Ms office, the following steps are required to access the Ms word : -

Starting MS-Word and creating Documents:-

1. Find the start button on the desktop and click on it once you have clicked on the start button the start button the start menu options.
2. Click over the program option of the start menu to open up a sub menu on left side of the start menu.
3. drag the mouse vertically down up to “Microsoft word” and click on it as shown.
4. First the MS-Word logo appears and start up window opens.

Opening New Documents:-

1. Open a new document by clicking file>new>ok.
2. In this new file type following matter word by word and taking care to type capital word as capital only.
3. The internet is a computer network made up of thousands of networks worldwide. No one knows exactly how many computer are connected to the network; although estimates are on going it is certain however, that these number in the millions and are increasing at a rapi role.
4. Save above document as sample 2.doc.

Open an existing file:-

1 start a Ms office application
2 On the menu bar click file
3 find and select open
4 specify the disk drive the directory name and the file name to open
5 click ok

Quick way :-

1 Start a Microsoft office application
2 on the tool bar click the open button
3 specify the disk drive the directory name and the file name to open
4 click ok.

Component of MS-Word 2007

Microsoft office button:-

The Microsoft button is locate at the upper left most corner and it contains a dop down shift of corners commands. Such a new open, save, save as, print, prepare, send publish and close. It also shows that the list of recently visited documents word option and exit word in the sub menu. The ESC key is used to move out of the activated drop down list. The option of the list can be activated action a user click on the option of the list.

MS Word Ribbon and Tabs

The Ribbon is a user interface element which was introduced by Microsoft in Microsoft Office. It is located below the Quick Access Toolbar and the Title Bar. It comprises seven tabs; Home, Insert, Page layout, References, Mailing, Review and View. Each tab has specific groups of related commands. It gives you quick access to the commonly used commands that you need to complete a task.

Home tab:

The Home tab is the default tab in Microsoft Word. It has five groups of related commands; Clipboard, Font, Paragraph, Styles and Editing. It helps you change document settings like font size, adding bullets, adjusting styles and many other common features. It also helps you to return to the home section of the document.

Clipboard Group

a) **Paste** This button is used to insert items that have been stored on the clipboard into a document.

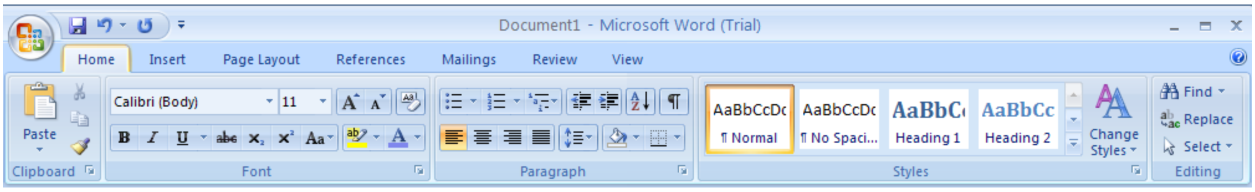
b) **Cut** To remove text or other items from a document and store it on the clipboard, click this button.

c) **Copy** Click this button to make a copy of selected text or another item in a document and store it on the clipboard.

d) **Format Painter** To make a copy of selected text and then apply the formatting to other text in a document, click this button. A single click will apply the formatting to other selected text one time. To apply the formatting to other selected text several times, click the format painter button twice. When all the formatting has been applied, click the button again to deactivate the format painter.

e) **Clipboard Launcher** This button is located in the lower right corner of the Clipboard group. It is used to open the Clipboard Task Pane. This is the task pane that will display when items that are added to the clipboard are displayed. This allows more than one item to be added to the clipboard before the items are pasted into the document.

Illustrate Others Yourself



Insert tab:

Insert Tab is the second tab in the Ribbon. As the name suggests, it is used to insert or add extra features in your document. It is commonly used to add tables, pictures, clip art, shapes, page number, etc. The Insert tab has seven groups of related commands; Pages, Tables, Illustrations, Links, Header & Footer, Text and Symbols.

a) Pages-Allows cover pages, blank pages, and page breaks to be added to the document.

b) Tables-Clicking the down arrow will open up a dialog box that will allow the user to insert tables, draw new tables, import Excel spreadsheets, and by clicking on the Quick Tables option (which opens up another dialog box) you can add a variety of pre-built items to the document.

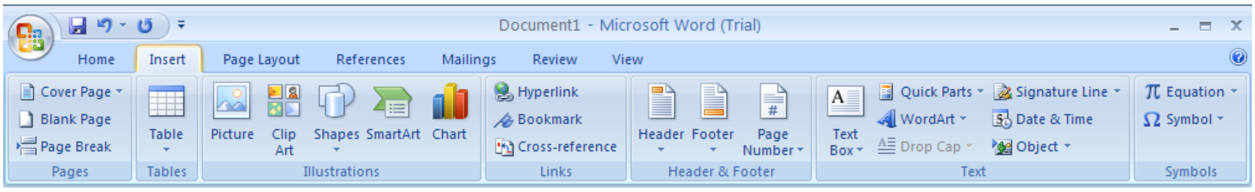
c) Illustrations-This group tab has options for adding images, clip art, charts, and more.

d) Links-The most commonly used item here is the Hyperlinks function, which allows or objects to be converted into URL links to addresses on the Internet.

e) Header & Footer-All three functions on this tab will open dialog boxes to give users many different options for headers, footers, and page numbering.

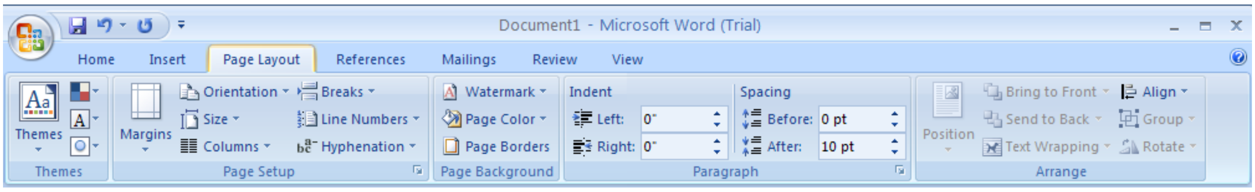
f) Text-This is not the same as using regular text in Word. Text boxes are objects that are in essence, mini-Word documents, and the other features offer text-like objects such as signatures, WordArt, and more.

g) Symbols-The two functions here, Equation and Symbol, both have down arrows that when clicked will open up dialog boxes showing a wide variety of equations and symbols that can be inserted into a document.



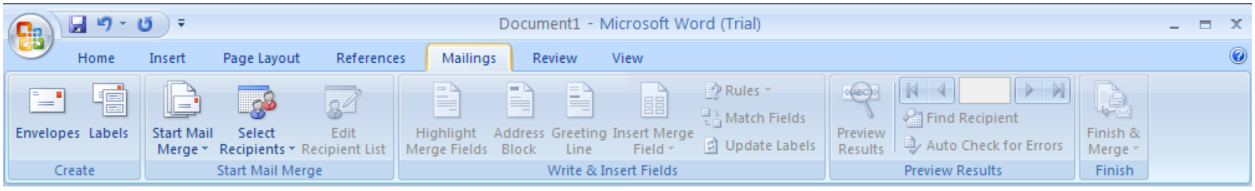
Page Layout tab:

It is the third tab in the Ribbon. This tab allows you to control the look and feel of your document, i.e. you can change the page size, margins, line spacing, indentation, documentation orientation, etc. The Page Layout tab has five groups of related commands; Themes, Page Setup, Page Background, Paragraph and Arrange.



Mailings tab:

It is the fifth tab in the ribbon. It is the least-often used tab of all the tabs available in the Ribbon. It allows you merge emails, writing and inserting different fields, preview results and convert a file into a PDF format. The Mailings tab has five groups of related commands; Create, Start Mail Merge, Write & Insert Fields, Preview Results and Finish.



Aim 5

Aim:- Creation and operating of spreadsheet using MS-Excel.

Worksheet

Worksheet is also called spreadsheet. It is made up of rows, columns and cells.

Rows

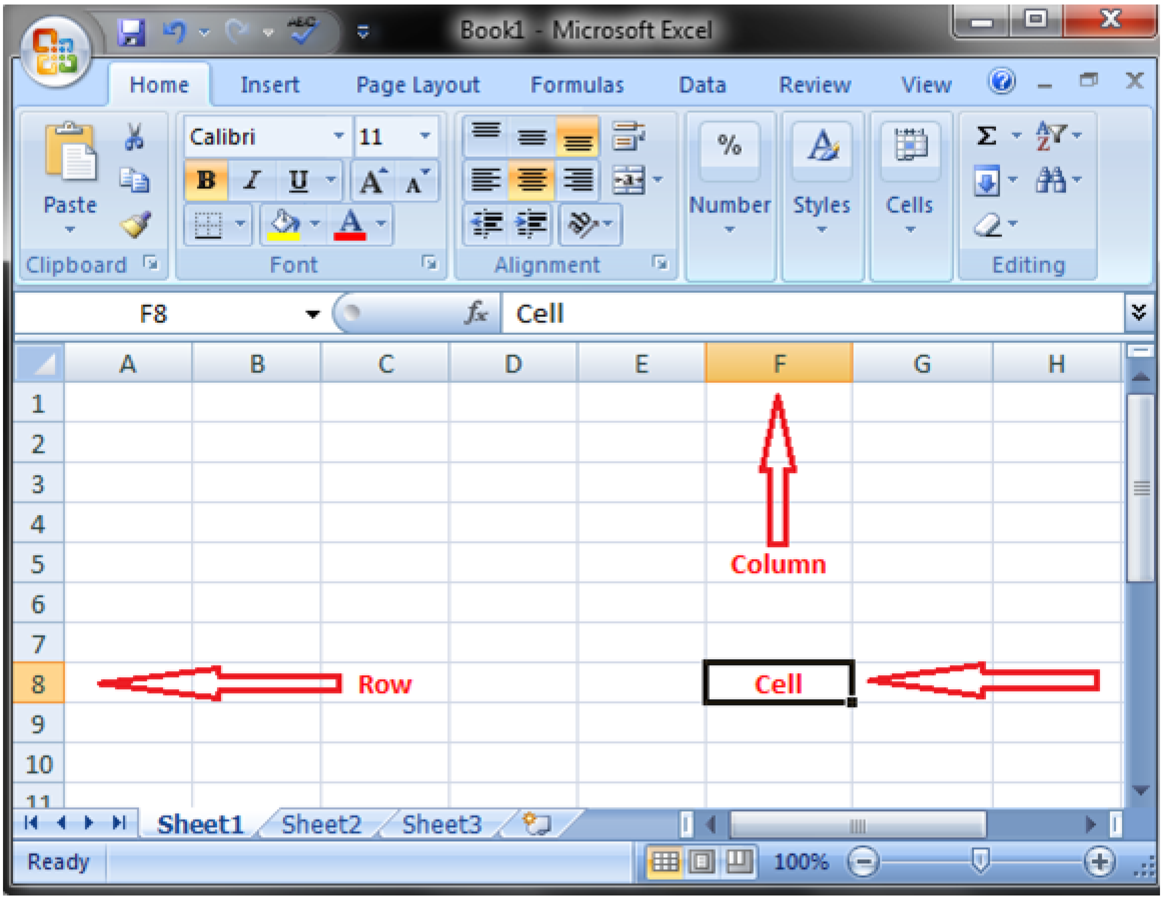
Rows run horizontally across the worksheet and ranges from 1 to 1048576. A row is identified by the number that is on left side of the row, from where the row originates.

Columns

Columns run vertically downward across the worksheet and ranges from A to XFD - 1 to 16384. A column is identified by a column header that is on the top of the column, from where the column originates.

Cells

Cells are small boxes in the worksheet where we enter data. A cell is the intersection of a row and column. It is identified by row number and column header such as A1, A2.



The features of the Insert tab are:

Tables, Illustrations, Charts, Links, Text

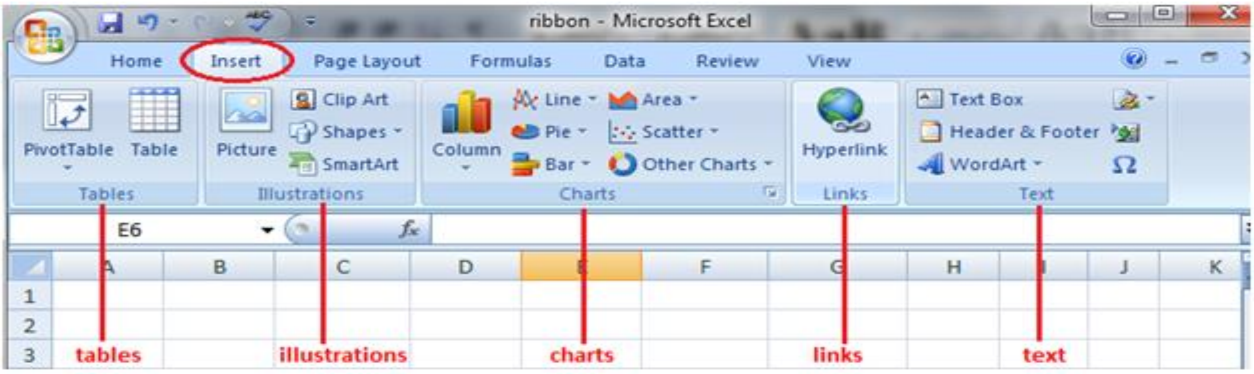
a) **Tables: -** We use this option to insert the dynamic table, Pivot table and recommended table. Pivot table is used to create the summary of report with the built-in calculation, and we have option to make our own calculation. Tables make it easy to sort, filter and format the data within a sheet. This option is also having recommended table that means on the basis of data, we can just insert the table as per the Excel's recommendation.

b) **Illustration: -** We use this option to insert the Pictures, Online Pictures, Shapes, SmartArt and Screenshot. It means if we want to insert any image, we can use Illustration feature.

c) **Charts: -**Charts is very important and useful function in Excel. In excel, we have different and good numbers of readymade chart options. We have 8 types of different charts in Excel:- Column, Bar, Radar, Line, Area, Combo, Pie and Bubbles chart. We can insert *Pivot chart* as well as Recommended chart, and if we don't know which chart we should insert for the data, we can use this option to fulfil the requirement.

d) **Links: -**We use this option to create the link in the document for the quick access to webpage and files.We can also use it to access different locations in the document.

e) **Text: -**We use this option to insert the Text box, Header and Footer, Word art, Signature and objects. We insert Text box to write something in the image format. We use Header and Footer options to place the content on the top and bottom of the page. Word art makes the text stylish. Insert the Add Signature Lines that specify the individual who is supposed to sign it. And object option works for embedded objects, like documents or other files we have inserted into the document.



The features in the Formula tab are:

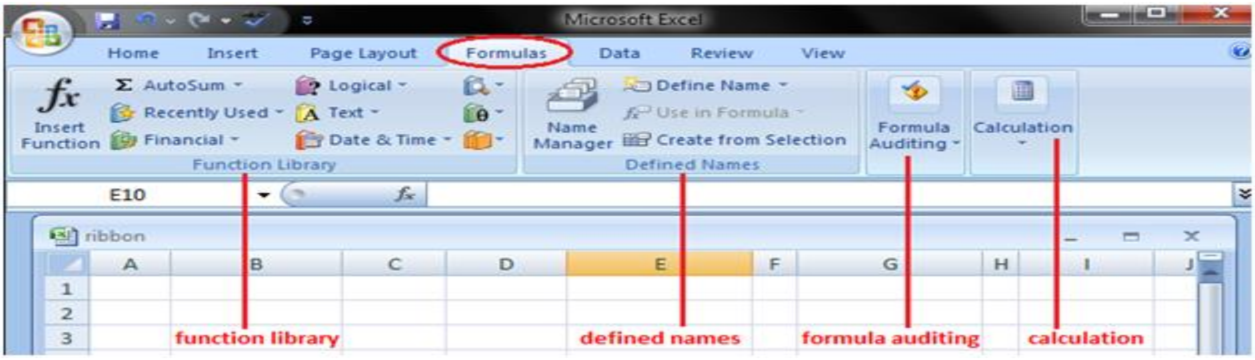
Function Library, Defined Names, Formula Auditing, Calculation

a) **Function Library: -**In Excel, we have 461 functions and they are available in the formulas tab under the function library group. And there is no need to learn every formula for we can find formulas category wise such as:- Financial, Logical, Text, Date & Time, Lookup & Reference, Math & Trig, many more.

b) **Defined Names: -** We use this option to define the name of a cell, range and the list of the already defined name can be viewed in Name manager and we can edit it if we want to change the range or cell reference of any defined name. Also, we can use the defined name in Formulas.

c) **Formula Auditing: -** This option helps us to identify the relation of formulas. By using Trace precedent, we get to know on which cell formula cell is based. Trace dependents are used to know if active cell is being used in any formula. When we use trace precedent and trace dependents the arrows get inserted automatically and, to remove the arrows, we use Remove Arrows. Show formula is a self-explained word; it is used to show the formula in the sheet. Error checking is used to check the error in the sheet. Evaluate formula option is used to evaluate the formula step by step.

d) **Calculation: -**If we want to switch the calculation from automatically to manually calculation and manually to automatically calculation, then we can do it by using the Calculation option. We use Calculate Now and Calculate Sheet option if automatic calculation is turned off.



The features in the Data tab are:

Get External Data, Connections, Sort & Filter, Data Tools and Outline

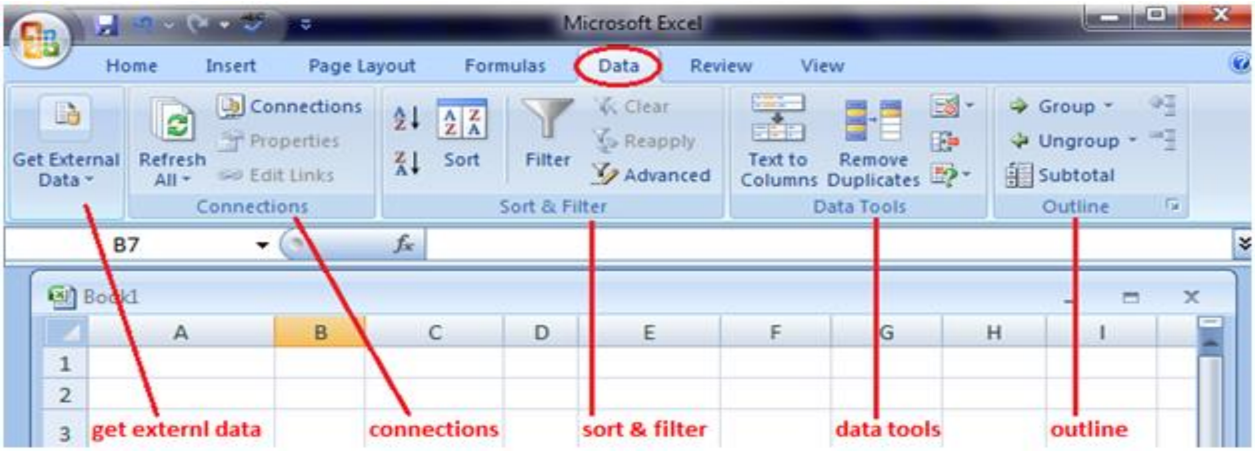
a) **Get External Data: -** In Excel, we can import data from MS Access, Web, Text and other sources. Also, we can import the data from other applications.

b) **Connections: -It is used to** display the entire data connections for the workbook. Data connections are links to the data outside the workbook which can be updated if the source data changes. And updated data can be obtained by refreshing all sources in workbook.

c) **Sort & Filter: -**To set the data in ascending or descending order on the basis of value or as per the first letter of a word, we use Sort option. Also, we can put the basic and advanced filter from here only.

d) **Data Tools: -**This option is very important to make the report interactive; it helps us to make the data authentic, and using this tab, we can restrict or validate the entries if the data is being updated by multiple users. Text to Columns helps us to split the single column into multiple columns as per data. Flash fills the values in the range. We can delete duplicate rows from the data by using Remove Duplicates option. We use Data Validation to provide the list that can be entered in cell, or we can restrict the entries, or we can validate the entries in the cell. We use Consolidate option to summarize data from separate ranges, and consolidate the result in a single output range. We use What-if-Analysis to analyse the data.

e) **Outline: -**We use this option to make the data more analytical and understandable. We can make group of rows or columns or automatically create an outline. We can ungroup the data; quickly calculate rows of related data by inserting subtotals and totals. We use Show and Hide options when we want to insert the Subtotal in data.

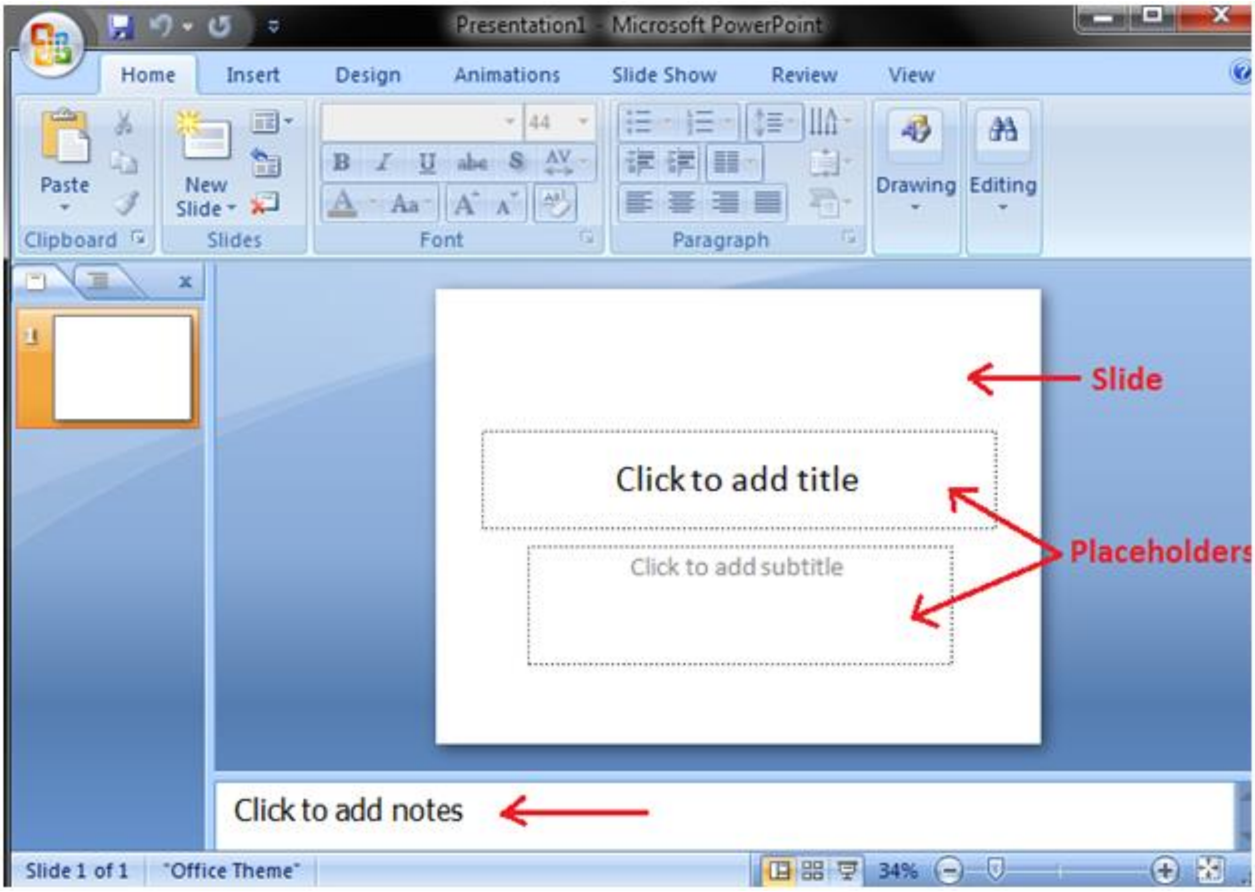


Aim 6

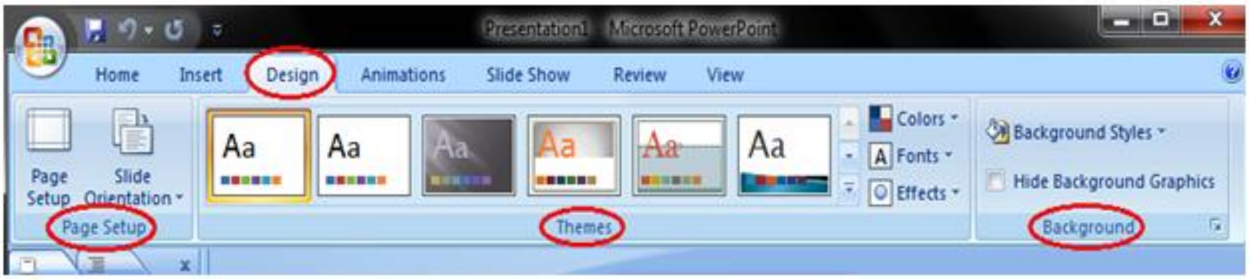
Aim:- Creation and editing power-point slides using **MS-** power point.

Theory:-

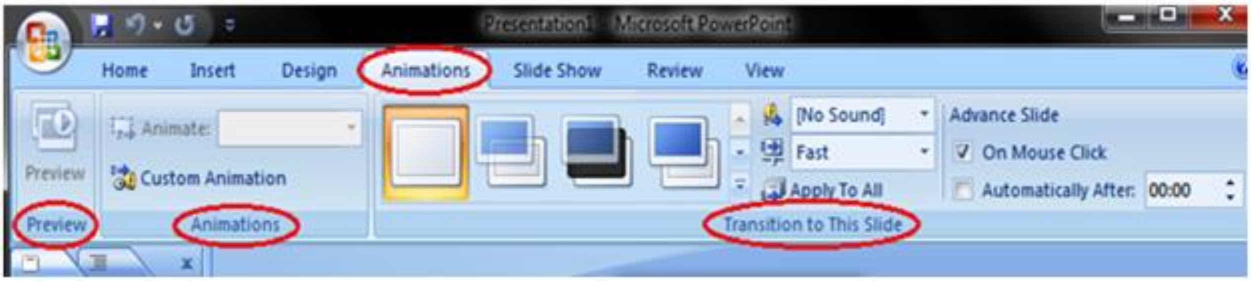
Slide: Presentation is created on slides. It lies in the centre of the PowerPoint window.
Placeholder: By default two placeholders appear in the slide when you open the PowerPoint.
Click to add notes: This space is provided to create notes if needed.



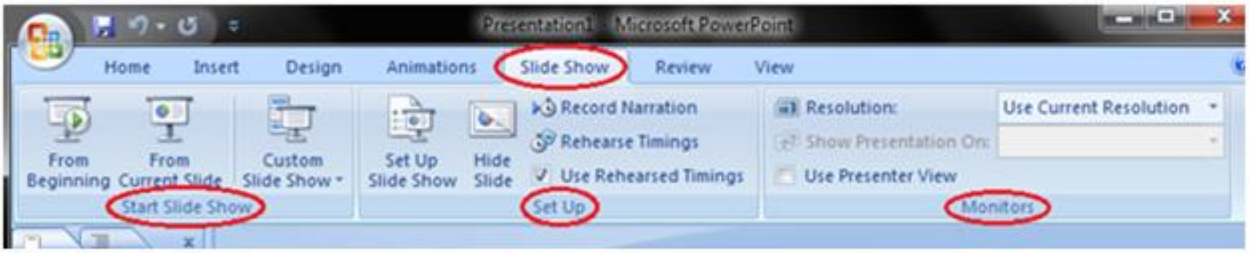
- **Design tab** – Enables the page setup and slide orientation (whether portrait or landscape). A rich variety of slide themes—with color templates, fonts, and effects—may be defined in this tab. Background styles and textures may be added as well. The features of Design tab are Page Setup, Themes and Background.



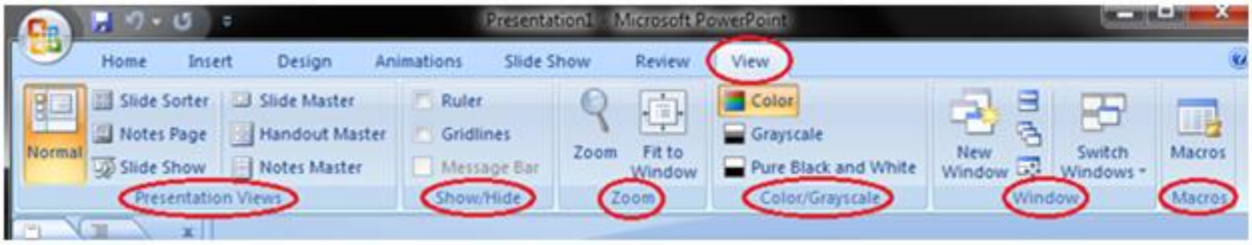
- **Animation tab** – Provides a variety of transition special effects, which are previewable with a simple mouseover of the particular effects. Transition sounds and speed also may be set here. The timing of slide advancements, whether manually based on mouse clicks or automated timings, may also be set. The features of Animations tab are Preview, Animations and Transition to This Slide.



- **Slide Show tab** – Provides a way to preview the slideshow and control its resolution and appearance.The features of Slide Show tab are Start Slide Show, Set Up and Monitors.



- **View tab** – Offers presentation views; the uses of rulers or gridlines for layout help; zoom features; color / grayscale effects; window views of the various slides; and downloaded or programmed macros for various effects.The features of View tab are Presentation Views, Show/Hide, Zoom, Color/Grayscale, Window and Macros.



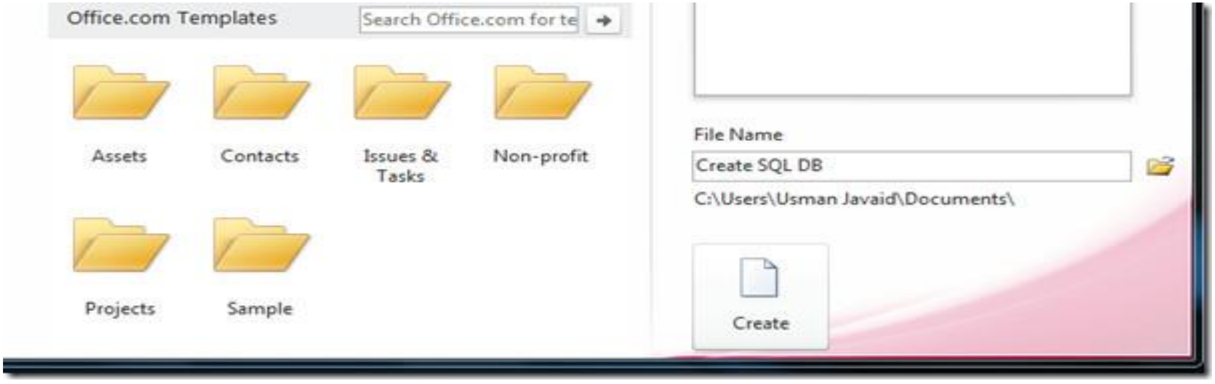
Aim 7

Aim:- Creation and manipulation of database table using SQL in MS-Access.

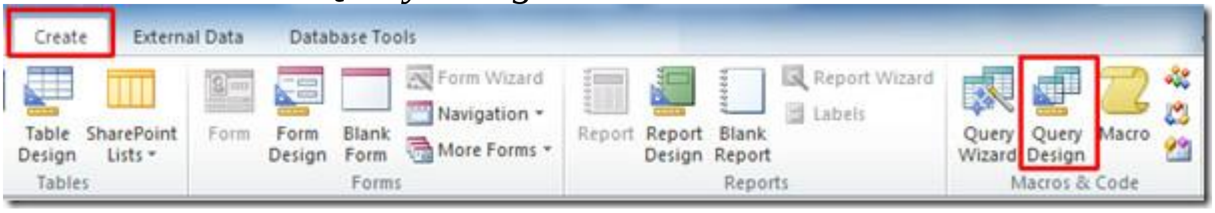
Theory:-

MS Access provides a number of ways to create tables and queries. Using simple SQL commands, you can create tables, queries, forms, and reports. SQL stands for Structured Query Language that is being used extensively for creating, handling and manipulating RDBMS. Its syntax is very easy to comprehend, but becomes complex when you try to handle the colossal database. In this post we will define a simple query with SQL commands, which eventually creates a simple table.

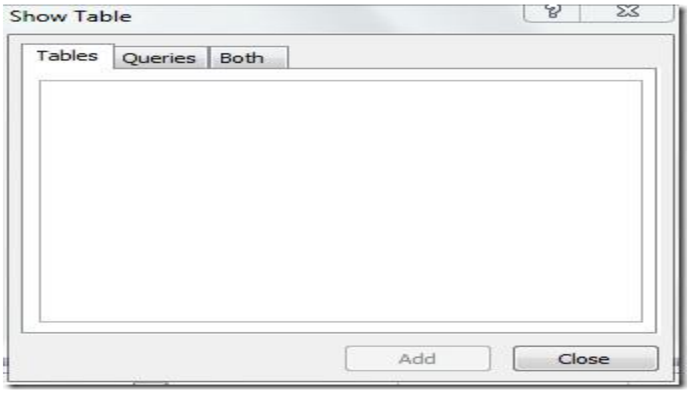
To start off with, Launch Access 2010 and give database an appropriate name. Once done, click *Create.*



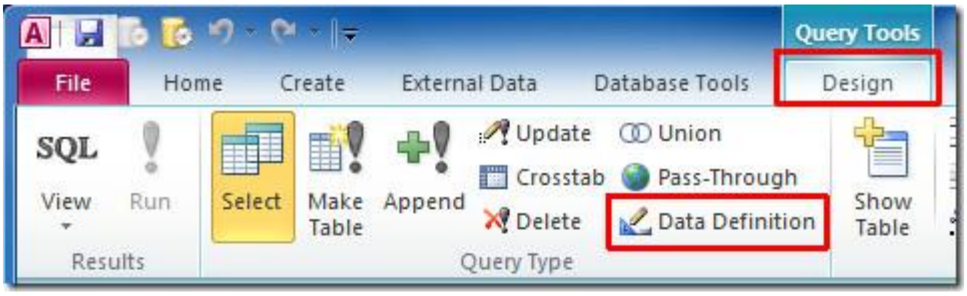
It will open a blank table, close it in order to remove it. Now, navigate to *Create* tab and click *Query Design*.



Upon click you will see *Show Table*, close this dialog box.



Now, head over to *Design* tab and click *Data Definition* from *Query Type* group.



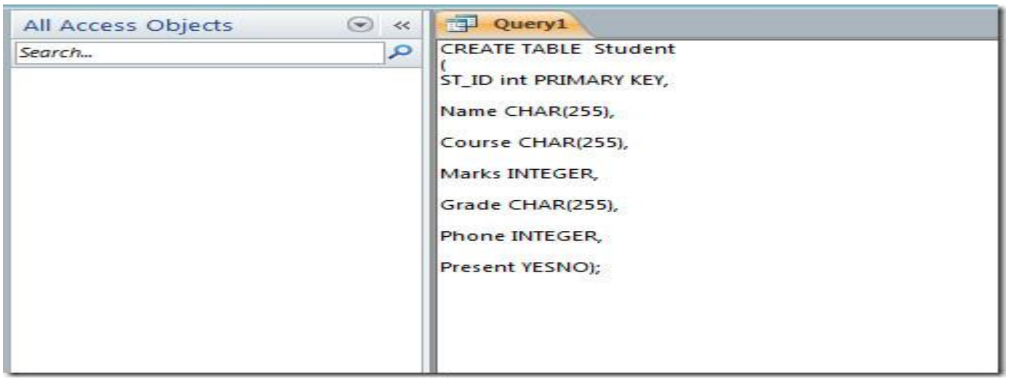
It will open *Query tab1*. Here, you can write SQL commands. Since we want to create simple table, we will write SQL commands to create table and important field labels with their respective data types.

CREATE TABLE Student
(
ST_ID int *PRIMARY KEY,*
Name CHAR(255),
Course CHAR(255),
Marks INTEGER,
Grade CHAR(255),
Phone INTEGER,

Present YESNO);

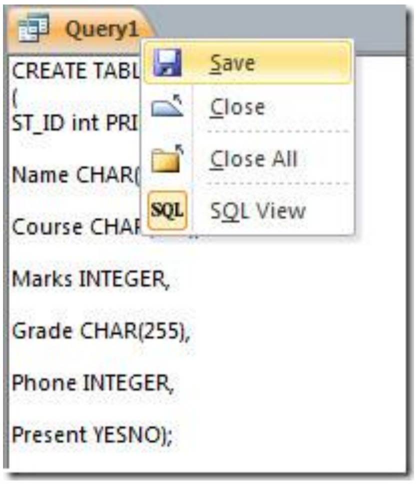
In SQL commands, we started off with *CREATE TABLE*, this will create a table with the name specified*Student.* The first field label *ST_ID* with data type int (Integer), we have also set it as PRIMARY KEY (unique identifier and non-repetitive). The next field label *Name* has data type CHAR(255) or *Text*. The last field label *Present* has data type *YESNO*, which confines your database users to enter either *Yes or NO.*

As you can see in SQL commands, we have enclosed field labeling & data type section in parenthesis and placed semicolon at the end. Syntax error would erupt if you don't comply with these standards.



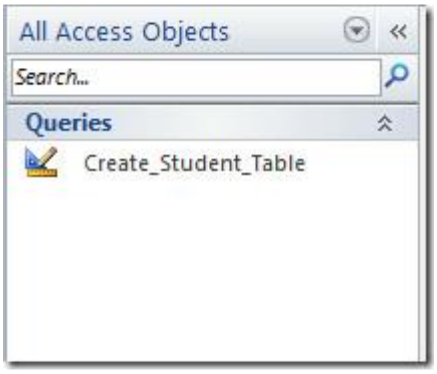
```
CREATE TABLE  Student
(
ST_ID int PRIMARY KEY,
Name CHAR(255),
Course CHAR(255),
Marks INTEGER,
Grade CHAR(255),
Phone INTEGER,
Present YESNO);
```

Now, save the query by right-clicking *Query1* tab and hit Save. Specify an appropriate name for it.

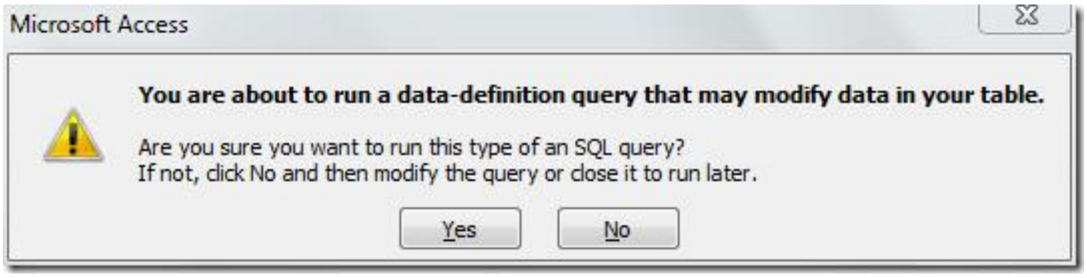


```
CREATE TABL
(
ST_ID int PRI
Name CHAR(
Course CHAR
Marks INTEGER,
Grade CHAR(255),
Phone INTEGER,
Present YESNO);
```

You will see the newly created query through *Data Definition* in the left sidebar of the database window.



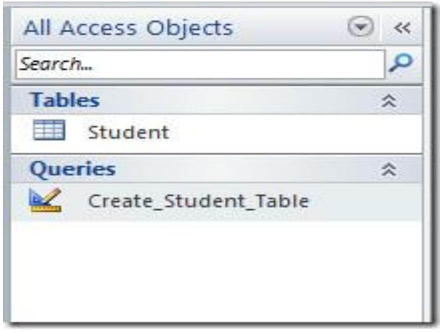
Now run this query, a message window will pop-up, asking you to confirm the action. Click *Yes* to continue.



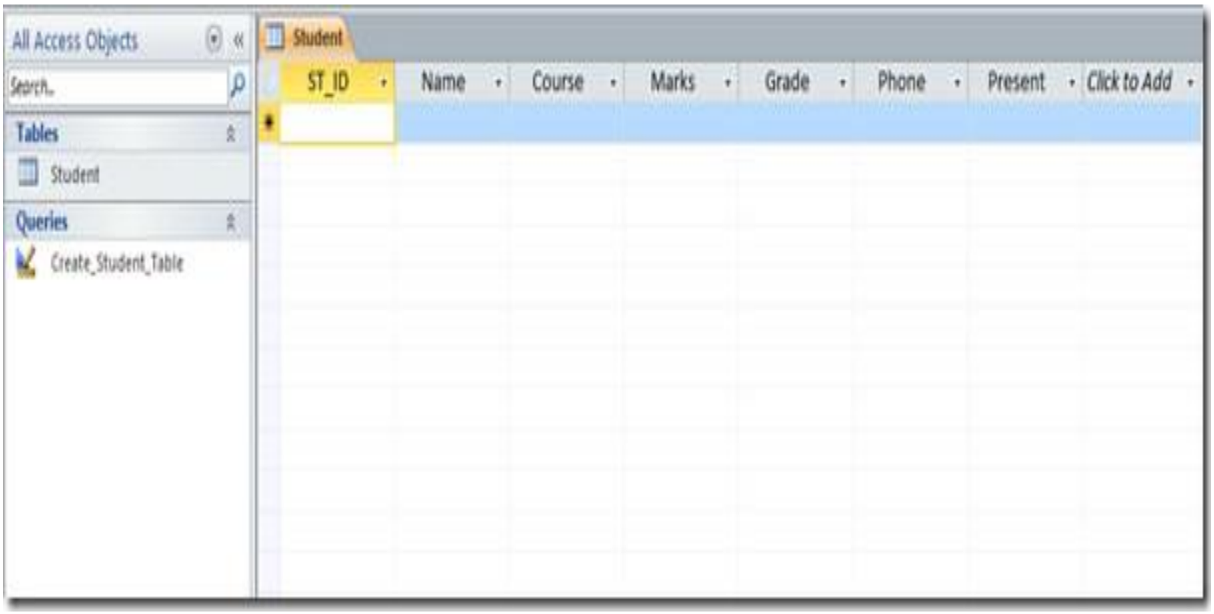
You are about to run a data-definition query that may modify data in your table.

Are you sure you want to run this type of an SQL query?
If not, click No and then modify the query or close it to run later.

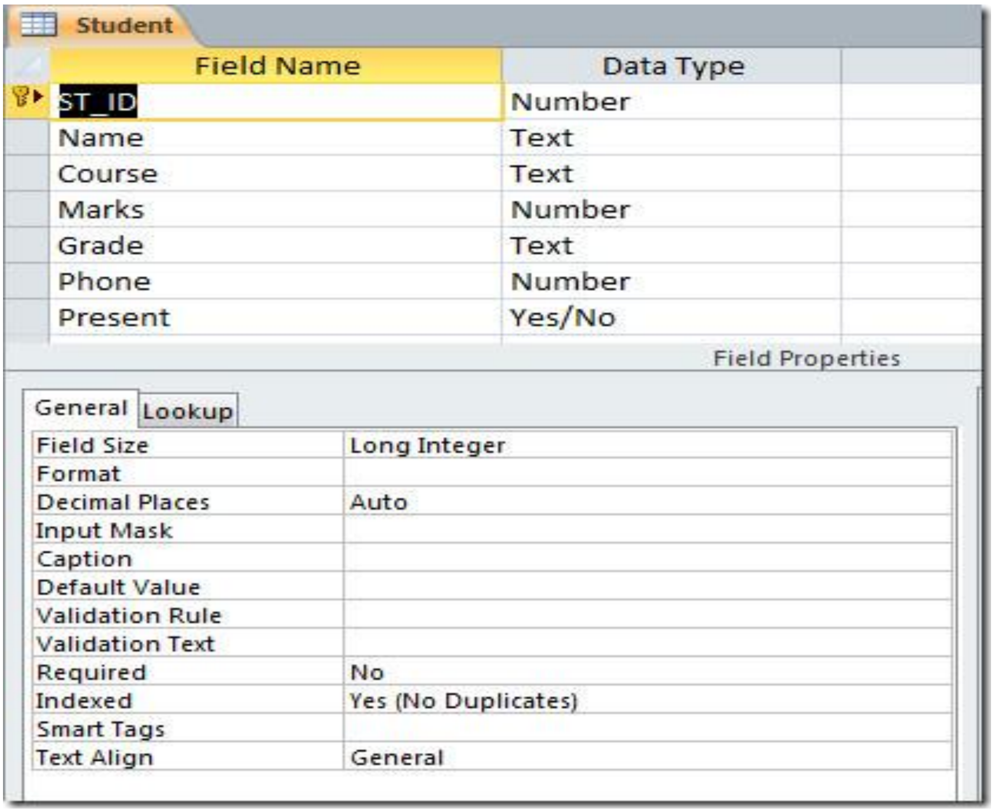
Upon click, it will do as specified in SQL command; it will create a table by the name of *Student*, as shown in the screenshot below.



On opening newly created table, you will see all the field labels as specified in SQL commands. As shown in the screenshot below that it has all the column labels; *ST_ID*, *Name*, *Course*, *Marks*, *Grade*, *Phone*, *and Present.*



Now switch to Design view and check the Data types. Here, you can apply new constraints for table data fields and add more field labels etc.



Field Name	Data Type
ST_ID	Number
Name	Text
Course	Text
Marks	Number
Grade	Text
Phone	Number
Present	Yes/No

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

Aim 8

Aim:- WAP to illustrate Arithematic expressions.

Algorithms:-

```
Step 1: Start
Step 2:input two integer variable let x,y
Step 3: add/sub/multi/division /Mod these two variables
Step 4: Store result in third variable
Step 5: Stop
```

Program code 1:-

```
#include <iostream.h>
using namespace std;
main() {
   int a = 21;
   int b = 10;
   int c ;
   c = a + b;
   cout << "Line 1 - Value of c is :" << c << endl ;

   c = a - b;
   cout << "Line 2 - Value of c is  :" << c << endl ;

   c = a * b;
   cout << "Line 3 - Value of c is :" << c << endl ;

   c = a / b;
   cout << "Line 4 - Value of c is  :" << c << endl ;

   c = a % b;
   cout << "Line 5 - Value of c is  :" << c << endl ;
   return 0;
}
```

output:-

```
Line 1 - Value of c is :31

Line 2 - Value of c is  :11

Line 3 - Value of c is :210

Line 4 - Value of c is  :2

Line 5 - Value of c is  :1
```

Aim 9

Aim:- WAP to illustrate Arrays.

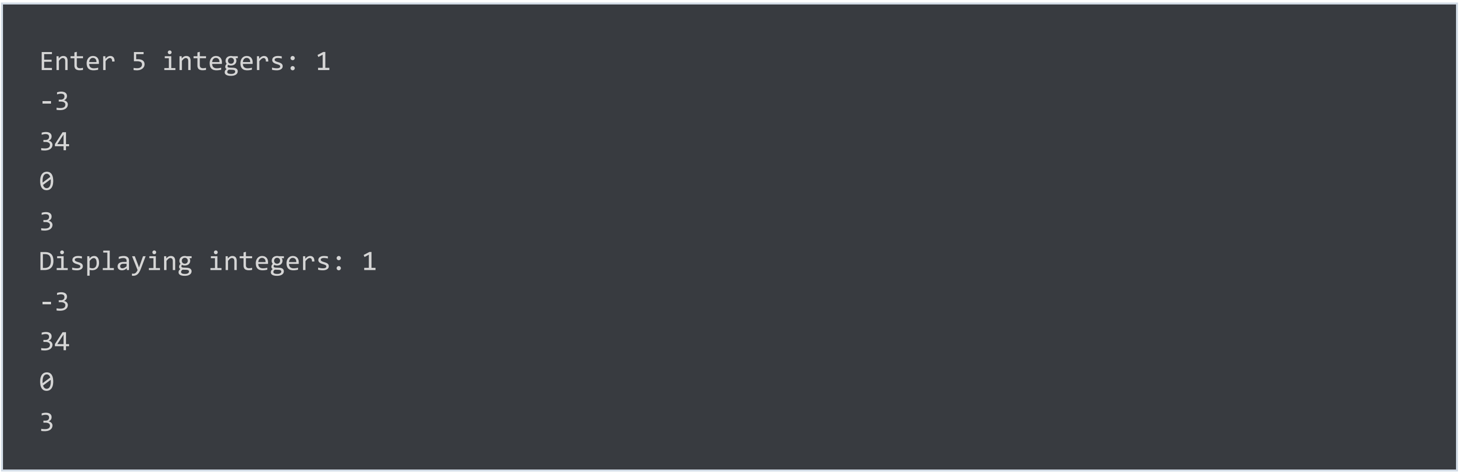
Algorithm:-

```
Step 1:- start
Step 2:- declare one integer array whose size is 5 like int values[5];
Step 3:- Assign the value of array with the help of For Loop.
            [for loop begin i=0 to 5 step 1]
             Set  n[i]=values
            Next
Step 4:- [End Loop]
Step 5:- Display the values of array with the help of For Loop.
            [for loop begin i=0 to 5 step 1]
            Print the values of array n
            Next
Step 6:- [End Loop]
Step 7:- End
```

Program code 9a:-

```
// Program to take 5 values from the user and store them in an array
// Print the elements stored in the array
#include <stdio.h>

int main() {
  int values[5];

  printf("Enter 5 integers: ");

  // taking input and storing it in an array
  for(int i = 0; i < 5; ++i) {
     scanf("%d", &values[i]);
  }
  printf("Displaying integers: ");

  // printing elements of an array
  for(int i = 0; i < 5; ++i) {
     printf("%d\n", values[i]);
  }
  return 0;
}
```

output:-



```
Enter 5 integers: 1
-3
34
0
3
Displaying integers: 1
-3
34
0
3
```

Program code 9b:-

```
#include <iostream.h>
#include<conio.h>
void main ()
{
   clrscr();
   char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
   cout << "Greeting message: ";
   cout << greeting << endl;
   getch();
}
```

output

```
Greeting message:Hello
```

Aim 10

Aim:- WAP to illustrate Functions.

Algorithm:-

```
Step 1:  start
Step 2:- declare one function like MAX which have integer return type & it take
         two integer values globally.
Step 3:- In main function
          Let three integer variable like a,b & ret
Step 4:-  Assign the values in variable a & b;
Step 5:-  Call function Max & passing the a & b as the argument & store the
          value in variable ret.
          Like set ret=MAX(a,b)
Step 6:-  Display the result which is stored in variable ret
Step 6:-  After the main function defination MAX function will be define.
Step 7:-  End
```

Program code 10a:-

```
#include <iostream.h>
using namespace std;

int max(int num1, int num2);   // function declaration

int main ()
{
   int a = 100;           // local variable declaration:
   int b = 200;
   int ret;
   ret = max(a, b);    // calling a function to get max value.
   cout << "Max value is : " << ret << endl;
    return 0;
}

int max(int num1, int num2)  // function returning the max between two
                                                      numbers
```

```
 {
     int result;                         // local variable declaration
    if (num1 > num2)
      result = num1;
   else
      result = num2;
      return result;
}
```

<u>Program code 10b:-</u>

```
#include <iostream.h>
#include<conio.h>
int add(int, int);          //Function prototype(declaration)
void main()
{   clrscr();
    int num1, num2, sum;
    cout<<"Enters two numbers to add: ";
    cin>>num1>>num2;
    sum = add(num1,num2);        //Function call
    cout<<"Sum = "<<sum;
    getch();
}

int add(int a,int b)             //Function defination
{
    int add;
    add = a+b;
    return add;                 //Return statement
}
```

<u>Output</u>

```
Enters two numbers to add:20
30
Sum =50
```

Aim 11

Aim:- WAP to illustrate constructor & Destructor.

Algorithm:-

```
Step 1:- start
Step 2:- declare one  class like student
          Let 4 variables one char type array, one integer type & two  float type.
          Let one default constructor, one display method which is no return no
          argument Type & one destructor.
Step 3:  both constructor & display method are define in public saction in class.
Step 4:   In main function
 Declare the object of class student, where as constructor automatic call & then
after display method call with the help of object & show the result destructor
will be call automatically.
Step 5:   end
```

Program code 11:-

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
private:
char name [25];
int roll;
float height, weight;
public:
student ()
{
strcpy (name, "Ram");
roll=0;
height=0;
weight=0;
}
void display ()
{
cout<<"\n Name :"<< name;
cout<<"\n Roll no"<<roll;
cout<<"\n Height"<<height;
cout<<"\n Weight"<<weight;
}
~student ()
{
cout<<"\n destroying object";
}
};
void main ()
{
student obj;
obj.student();
getch ();
}
```

Output

```
 Name : Ram
Roll no : 0
Heigth:0
Weight:0
```

Aim 12

Aim:- WAP to illustrate Object & classes.

Algorithm:-

```
Step 1: Start.
Step 2: Declear One Class Like Box.
          Let 2 VariableLength And Breadth.
          Let One Method Area Which is no return no Argument type.
Step 3:Method Area Define Outside The class.
Step 4:- In Main Function
           Declare One Object Of Class Box.
Step 5:- The Help of Object Called Area Method And Show The Result.
Step 6:- End.
```

Program code 12:-

```
#include <iostream>

using namespace std;

class Box {
   private:
      double length;   // Length of a box
      double breadth;  // Breadth of a box
      public:
       void area();
};
Void box::area()
{
Cout<<"Enter the length & breadth";
Cin>>length>>breadth;
Cout<<endl<<"Result is :"<<length*breadth;
}
void main() {
   Box Box1;        // Declare Box1 of type Box
   Box1.area();     }
```

output:-

```
Enter the length & breadth : 21
30
Result is : 630
```

Aim 13

Aim:- WAP to illustrate operator overloading.

Algorithm:-

```
Step 1: Start the program.
Step 2: Declare the class.
Step 3: Declare the variables and its member function.
Step 4: Using the function getvalue() to get the two numbers.
Step 5: Define the function operator +() to add two complex numbers.
Step 6: Define the function operator –()to subtract two complex numbers.
Step 7: Define the display function.
Step 8: Declare the class objects obj1,obj2 and result.
Step 9: Call the function getvalue using obj1 and obj2
Step 10: Calculate the value for the object result by calling the function
         operator + and  operator -.
Step 11: Call the display function using obj1 and obj2 and result.
Step 12: Return the values.
Step 13: Stop the program.
```

Program code 13:-

```
#include<iostream.h>
#include<conio.h>
class complex {
   int a, b;
public:
   void getvalue() {
      cout << "Enter the value of Complex Numbers a,b:";
      cin >> a>>b;
   }
   complex operator+(complex ob) {
      complex t;
      t.a = a + ob.a;
      t.b = b + ob.b;
      return (t);
   }
   complex operator-(complex ob) {
      complex t;
      t.a = a - ob.a;
```

```
        t.b = b - ob.b;
        return (t);
    }
    void display() {
        cout << a << "+" << b << "i" << "\n";
    }
};
void main() {
    clrscr();
    complex obj1, obj2, result, result1;
    obj1.getvalue();
    obj2.getvalue();
    result = obj1 + obj2;
    result1 = obj1 - obj2;
    cout << "Input Values:\n";
    obj1.display();
    obj2.display();
    cout << "Result:";
    result.display();
    result1.display();
    getch();
}
```

output:-

```
Enter the value of Complex Numbers a, b

4              5

Enter the value of Complex Numbers a, b

2              2

Input Values

4 + 5i

2 + 2i

Result

6 +   7i

2 +   3i
```

Aim 14

Aim:- WAP to illustrate Function overloading.

Algorithm:-

```
STEP 1:  Start the program.
STEP 2:  Declare the class name as fn with data members and member
             functions.
STEP 3:  Read the choice from the user.
STEP 4:  Choice=1 then go to the step 5.
STEP 5:  The function area() to find area of circle with one integer argument.
STEP 6:  Choice=2 then go to the step 7.
STEP 7:  The function area() to find area of rectangle with two integer
           argument.
STEP 8:  Choice=3 then go to the step 9.
STEP 9:  The function area() to find area of triangle with three arguments, two
           as Integer and one as float.
STEP 10: Choice=4 then stop the program.
```

Program code 14:-

```
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
#define pi 3.14
class fn {
public:
   void area(int); //circle
   void area(int, int); //rectangle
   void area(float, int, int); //triangle
};

void fn::area(int a) {

   cout << "Area of Circle:" << pi * a*a;
}

void fn::area(int a, int b) {
   cout << "Area of rectangle:" << a*b;
}
```

```
void fn::area(float t, int a, int b) {
   cout << "Area of triangle:" << t * a*b;
}

void main() {
   int ch;
   int a, b, r;
   clrscr();
   fn obj;
   cout << "\n\t\tFunction Overloading";
   cout << "\n1.Area of Circle\n2.Area of Rectangle\n3.Area of
Triangle\n4.Exit\n:?;
         cout << ?Enter your Choice : ";
         cin>>ch;

   switch (ch) {
      case 1:
         cout << "Enter Radious of the Circle:";
         cin>>r;
         obj.area(r);
         break;
      case 2:
         cout << "Enter Sides of the Rectangle:";
         cin >> a>>b;
         obj.area(a, b);
         break;
      case 3:
         cout << "Enter Sides of the Triangle:";
         cin >> a>>b;
         obj.area(0.5, a, b);
         break;
      case 4:
         exit(0);
   }
   getch();
}
```

output:-

```
Function Overloading
1. Area of Circle
2. Area of Rectangle
3. Area of Triangle
4. Exit
Enter Your Choice: 2

Enter the Sides of the Rectangle: 5 5

Area of Rectangle is: 25

1. Area of Circle
2. Area of Rectangle
3. Area of Triangle
4. Exit
Enter Your Choice: 4
```

Aim 15

Aim:- WAP to illustrate Derived classes & Inheritance.

Algorithm:-

```
Step 1: Start the program.
Step 2: Declare the base class emp.
Step 3: Define and declare the function get() to get the employee
details.
Step 4: Declare the derived class salary.
Step 5: Declare and define the function get1() to get the salary details.
Step 6: Define the function calculate() to find the net pay.
Step 7: Define the function display().
Step 8: Create the derived class object.
Step 9: Read the number of employees.
Step 10: Call the function get(),get1() and calculate() to each
employees.
Step 11: Call the display().
Step 12: Stop the program.
```

Program code 15:-

```
#include<iostream.h>
#include<conio.h>

class emp {
public:
   int eno;
   char name[20], des[20];

   void get() {
      cout << "Enter the employee number:";
      cin>>eno;
      cout << "Enter the employee name:";
      cin>>name;
      cout << "Enter the designation:";
      cin>>des;
   }
};
```

```
class salary : public emp {
    float bp, hra, da, pf, np;
public:

    void get1() {
        cout << "Enter the basic pay:";
        cin>>bp;
        cout << "Enter the Humen Resource Allowance:";
        cin>>hra;
        cout << "Enter the Dearness Allowance :";
        cin>>da;
        cout << "Enter the Profitablity Fund:";
        cin>>pf;
    }

    void calculate() {
        np = bp + hra + da - pf;
    }

    void display() {
        cout << eno << "\t" << name << "\t" << des << "\t" << bp << "\t" << hra
<< "\t" << da << "\t" << pf << "\t" << np << "\n";
    }
};

void main() {
    int i, n;
    char ch;
    salary s[10];
    clrscr();
    cout << "Enter the number of employee:";
    cin>>n;
    for (i = 0; i < n; i++) {
        s[i].get();
        s[i].get1();
        s[i].calculate();
    }
    cout << "\ne_no \t e_name\t des \t bp \t hra \t da \t pf \t np \n";
    for (i = 0; i < n; i++) {
        s[i].display();
    }
    getch();
}
```

output:-

```
Enter the Roll no: 100
Enter two marks

90
80

Enter the Sports Mark: 90

Roll No: 100
Total    : 260
Average: 86.66
```

Aim 16

Aim:- WAP to insert and delete and element from the Stack.

Algorithm:-

Insert (Push) element in a Stack.

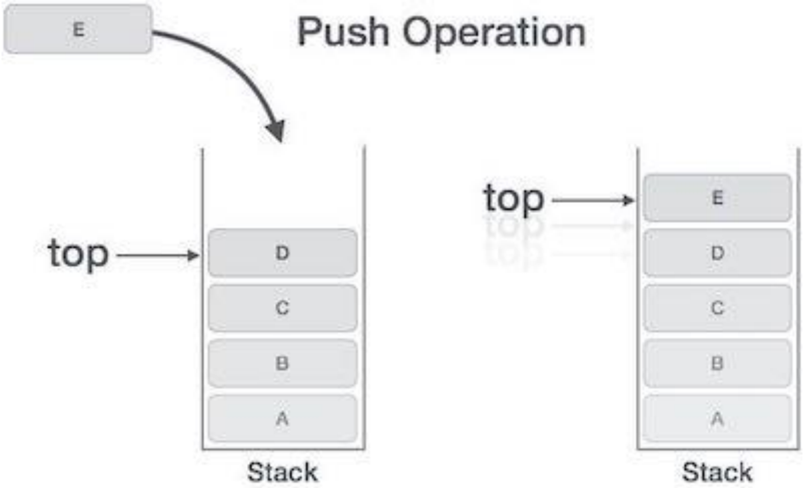
Step 1 − Checks if the stack is full.

Step 2 − If the stack is full, produces an error and exit.

Step 3 − If the stack is not full, increments **top** to point next empty space.

Step 4 − Adds data element to the stack location, where top is pointing.

Step 5 − Returns success.



Algorithm:- **Delete (POP) element From the Stack**

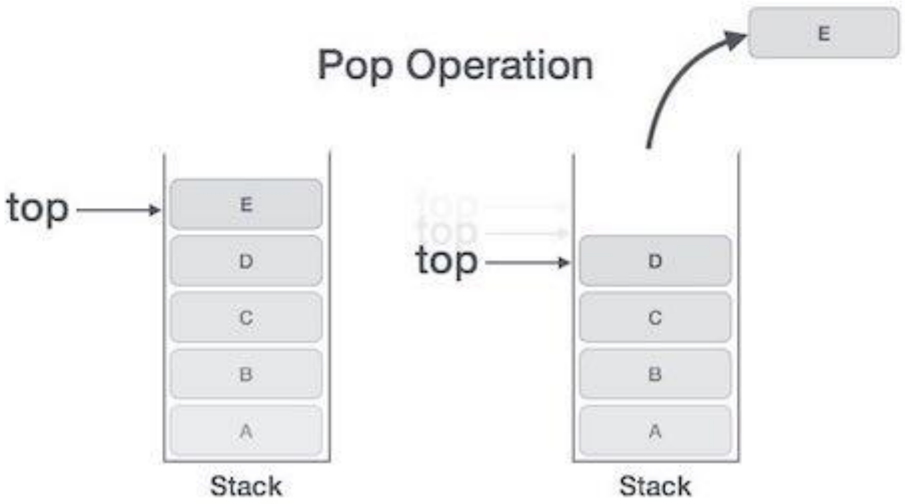
Step 1 − Checks if the stack is empty.

Step 2 − If the stack is empty, produces an error and exit.

Step 3 − If the stack is not empty, accesses the data element at which **top** is pointing.

Step 4 − Decreases the value of top by 1.

Step 5 − Returns success.



<u>Program code 16</u>:- /* C++ program to implement basic stack operations */

```
#include <bits/stdc++.h>

using namespace std;

#define MAX 1000

class Stack {
   int top;

public:
   int a[MAX]; // Maximum size of Stack

   Stack() { top = -1; }
   bool push(int x);
   int pop();
   bool isEmpty();
};

bool Stack::push(int x)
{
   if (top >= (MAX - 1)) {
      cout << "Stack Overflow";
      return false;
   }
   else {
        a[++top] = x;
        cout << x << " pushed into stack\n";
        return true;
    }
}

int Stack::pop()
{
    if (top < 0) {
        cout << "Stack Underflow";
        return 0;
    }
    else {
        int x = a[top--];
        return x;
    }
}

bool Stack::isEmpty()
{
    return (top < 0);
}

// Driver program to test above functions
int main()
{
    struct Stack s;
    s.push(10);
    s.push(20);
    s.push(30);
    cout << s.pop() << " Popped from stack\n";

    return 0;
}
```

output:-

```
10 pushed into stack

20 pushed into stack

30 pushed into stack

30 popped from stack
```

Aim 17

Aim:- WAP to insert and delete and element from the Queue.

Algorithm :-

Insert element from the Queue.

```
 Step 1: If REAR >= SIZE – 1 then
           Write "Queue is Overflow"
Step 2: REAR = REAR + 1
Step 3: QUEUE [REAR] = X
Step 4: If FRONT = -1 then
           FRONT = 0
```

Algorithm :-

Delete element from the Queue.

```
Step 1: If FRONT = -1 then
          Write "Queue is Underflow"
Step 2: Return QUEUE [FRONT]
Step 3: If FRONT = REAR then
          FRONT = 0
          REAR = 0
          Else
          FRONT = FRONT + 1
```

Program code 17:-

```
#include <stdio.h>
#define MAX 50
int queue_array[MAX];
int rear = - 1;
int front = - 1;
main()
{
   int choice;
   while (1)
   {
      printf("1.Insert element to queue \n");
      printf("2.Delete element from queue \n");
      printf("3.Display all elements of queue \n");
```

```
        printf("4.Quit \n");
        printf("Enter your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
           case 1:
           insert();
           break;
           case 2:
           delete();
           break;
           case 3:
           display();
           break;
           case 4:
           exit(1);
           default:
           printf("Wrong choice \n");
        }        /*End of switch*/
    }            /*End of while*/
}           /*End of main()*/
insert()
{
    int add_item;
    if (rear == MAX - 1)
    printf("Queue Overflow \n");
    else
    {
        if (front == - 1)
                      /*If queue is initially empty */
        front = 0;
        printf("Inset the element in queue : ");
        scanf("%d", &add_item);
        rear = rear + 1;
        queue_array[rear] = add_item;
    }
}              /*End of insert()*/

delete()
{
   if (front == - 1 || front > rear)
   {
      printf("Queue Underflow \n");
      return ;
   }
   else
   {
      printf("Element deleted from queue is : %d\n", queue_array[front]);
      front = front + 1;
   }
}           /*End of delete() */
display()
{
   int i;
   if (front == - 1)
      printf("Queue is empty \n");
   else
   {
      printf("Queue is : \n");
      for (i = front; i <= rear; i++)
         printf("%d ", queue_array[i]);
      printf("\n");
   }
}          /*End of display() */
```

output:-

```
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 10
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 15
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 20
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 30
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Element deleted from queue is : 10
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is :
15 20 30
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 4
```

Aim 18

Aim:- WAP to insert and delete and element from the Linked List.

Algorithm:-

Step 1 Push data in linked list Stack top position
Step 2 Pop data in linked list Stack based top position
Step 3 Display Nodes in Linked List Stack
Step 4 Count Nodes in Linked List Stack

Program code 18:-// C++ program for insertion in a single linked // list at a specified position.

```
#include <bits/stdc++.h>
using namespace std;

// A linked list Node
struct Node {
   int data;
   struct Node* next;
};

// Size of linked list
int size = 0;

// function to create and return a Node
Node* getNode(int data)
{
   // allocating space
   Node* newNode = new Node();

   // inserting the required data
   newNode->data = data;
   newNode->next = NULL;
   return newNode;
}

// function to insert a Node at required postion
void insertPos(Node** current, int pos, int data)
{
   // This condition to check whether the
```

```
    // postion given is valid or not.
    if (pos < 1 || pos > size + 1)
        cout << "Invalid postion!" << endl;
    else {

        // Keep looping until the pos is zero
        while (pos--) {

            if (pos == 0) {

                // adding Node at required postion
                Node* temp = getNode(data);

                // Making the new Node to point to
                // the old Node at the same position
                temp->next = *current;

                // Changing the pointer of the Node previous
                // to the old Node to point to the new Node
                *current = temp;
            }
            else
              // Assign double pointer variable to point to the
              // pointer pointing to the address of next Node
              current = &(*current)->next;
        }
        size++;
    }
}

// This function prints contents
// of the linked list
void printList(struct Node* head)
{
    while (head != NULL) {
        cout << " " << head->data;
        head = head->next;
    }
    cout << endl;
}
```

```
// Driver Code
int main()
{
    // Creating the list 3->5->8->10
    Node* head = NULL;
    head = getNode(3);
    head->next = getNode(5);
    head->next->next = getNode(8);
    head->next->next->next = getNode(10);

    size = 4;

    cout << "Linked list before insertion: ";
    printList(head);

    int data = 12, pos = 3;
    insertPos(&head, pos, data);
    cout << "Linked list after insertion of 12 at position 3: ";
    printList(head);
      // front of the linked list
    data = 1, pos = 1;
    insertPos(&head, pos, data);
    cout << "Linked list after insertion of 1 at position 1: ";
    printList(head);
      // insetion at end of the linked list
    data = 15, pos = 7;
    insertPos(&head, pos, data);
    cout << "Linked list after insertion of 15 at position 7: ";
    printList(head);

    return 0;
}
```

output:-

```
Linked list before insertion:  3 5 8 10

Linked list after insertion of 12 at position 3:  3 5 12 8 10

Linked list after insertion of 1 at position 1:  1 3 5 12 8 10

Linked list after insertion of 15 at position 7:  1 3 5 12 8 10 15
```

Enhancement by GGITS

Aim 19

Aim:- WAP to implement Virtual function.

Algorithm:-

```
Step 1: Start the program.
Step 2: Declare the base class base.
Step 3: Declare and define the virtual function show().
Step 4: Declare and define the function display().
Step 5: Create the derived class from the base class.
Step 6: Declare and define the functions display() and show().
Step 7: Create the base class object and pointer variable.
Step 8: Call the functions display() and show() using the base class object and
            pointer.
Step 9: Create the derived class object and call the functions display() and
          show() using the derived class object and pointer.
Step 10: Stop the program
```

Program code 19a:-

```
#include<iostream.h>
#include<conio.h>

class base {
public:
   virtual void show() {
      cout << "\n  Base class show:";
   }

   void display() {
      cout << "\n  Base class display:";
   }
};

class drive : public base {
public:

   void display() {
      cout << "\n  Drive class display:";
   }

   void show() {
      cout << "\n  Drive class show:";
   }
};

void main() {
   clrscr();
   base obj1;
   base *p;
   cout << "\n\t P points to base:\n";

   p = &obj1;
   p->display();
   p->show();

   cout << "\n\n\t P points to drive:\n";
   drive obj2;
   p = &obj2;
   p->display();
   p->show();
   getch();
}
```

output:-

```
P points to Base


Base class display
Base class show


P points to Drive


Base class Display
Drive class Show
```

Aim 20

Aim:- WAP to implement Friend Function.

Algorithm:-

```
STEP 1:  Start the program.
STEP 2:  Declare the class name as Base with data members and member
            functions.
STEP 3:  The function get() is used to read the 2 inputs from the user.
STEP 4:  Declare the friend function mean(base ob) inside the class.
STEP 5:  Outside the class to define the friend function and do the following.
STEP 6:  Return the mean value (ob.val1+ob.val2)/2 as a float.
STEP 7:  Stop the program.
```

Program code 20:-

```
#include<iostream.h>
#include<conio.h>

class base {
    int val1, val2;
public:
    void get() {
        cout << "Enter two values:";
        cin >> val1>>val2;
    }
    friend float mean(base ob);
};

float mean(base ob) {
    return float(ob.val1 + ob.val2) / 2;
}
void main() {
    clrscr();
    base obj;
    obj.get();
    cout << "\n Mean value is : " << mean(obj);
    getch();
}
```

output:-

```
Enter two values: 10, 20

Mean Value is: 15
```