INSTRUCTION MANUAL

FUNDAMENTALS OF COMPUTER PROGRAMMING
&
INFORMATION TECHNOLOGY
(BTCS-102)

Prepared by

Er. Kamaldeep Kaur
Assistant Professor (CSE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE
LUDHIANA – 141006

DECLARATION

This Manual of Fundamentals of Computer Programming & Information Technology (BTCS-102) has been prepared by me as per syllabus of Fundamentals of Computer programming & Information Technology (BTCS-102).

Signature

SYLLABUS

FUNDAMENTALS OF COMPUTER PROGRAMMING
&
INFORMATION TECHNOLOGY
(BTCS – 102)

External Marks: 20 **L T P**

Internal Marks: 30 **- - 2**

Total Marks: 50

1. Familiarization with the Computer System.

2. Navigating with Window Explorer, Working with Control Panel, to work at the command prompt, to understand the working of virus guards and antivirus software, exploring the Internet.

3. Word Processor, Spreadsheets, Presentation software.

4. Write a program to find the nature of the roots as well as value of the roots. However, in case of imaginary roots, find the real part and imaginary part separately.

5. Write a program, which takes two integer operands and one operator form user, performs the operation and then prints the result. (Consider the operators +,-,*, / % and use switch statement).For example, the input should be in the form: 5 + 3 the output should comes Result = 8

6. Fibonacci sequence is defined as follows: the first and second terms in the

sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a program to generate the first n terms of the sequence. For example, for n = 8, the output should be 0 1 1 2 3 5 8
13

7. Write a program to print all the prime numbers between m and n, where the value of m and n is supplied by the user.

8. The number such as 1991, is a palindrome because it is same number when read forward or backward. Write a program to check whether the given number is palindrome or not.

9. A positive integer number IJK is said to be well-ordered if I<J<K. For example, number 138 is called well-ordered because the digits in the number (1, 3, 8) increase from left to right, i.e., 1 < 3 < 8. Number 365 is not well-ordered because 6 is larger than 5. Write a program that will find and display all possible three digit well-ordered numbers. The program should also display the total number of three digit well-ordered numbers found.

10. Write a function to computer the highest common factor of integer numbers m and n. Use this function to find the highest common factor of integer numbers a and b.

11. Given the marks (out of 100) obtained by each student in a test of a class with n students. Write a program to obtain the following information:

(a) minimum and maximum marks score

(b) average score of the class, and

(c) number of students whose score is greater than class's average score

12. Write a program to multiply matrix Am×n by Bp×q, given that n = p.

13. Write a program to sort a list of n integer numbers in descending order using bubble sort method.

14. Write a program to perform the addition and multiplication of two complex numbers using structures

15. Write a program using pointers to compute the sum of all elements stored in an array.

Note: Students are required to prepare a file containing lab exercises based on programming only, where as the oral examination will from the entire syllabus.

INDEX

PRACTICAL 1

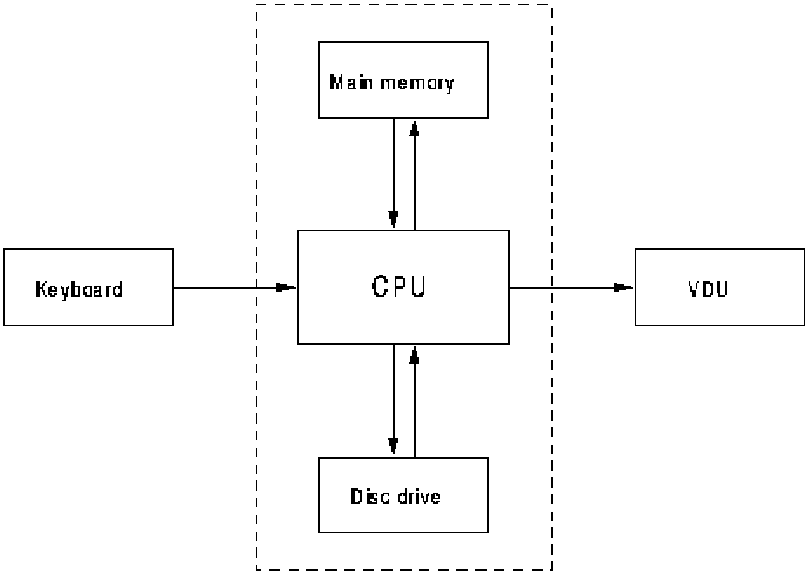
Aim: Familiarization with the Computer System

Theory:

Technically, a computer is a programmable machine. This means it can execute a programmed list of instructions and respond to new instructions that it is given. Today, however, the term is most often used to refer to the desktop and laptop computers that most people use. When referring to a desktop model, the term "computer" technically only refers to the computer itself -- not the monitor, keyboard, and mouse.

Some of the major parts of a personal computer (or PC) include the motherboard, CPU, memory (or RAM), hard drive, and video card. While personal computers are by far the most common type of computers today, there are several other types of computers. For example, a "minicomputer" is a powerful computer that can support many users at once. A "mainframe" is a large, high-powered computer that can perform billions of calculations from multiple sources at one time. Finally, a "supercomputer" is a machine that can process billions of instructions a second and is used to calculate extremely complex calculations

The following schematic diagram gives the layout of a Personal Computer (PC), most single user systems follow this general design:



The components perform the following tasks:

CPU (Central Processor Unit) -- does the `work', fetches, stores and manipulates values that are stored in the computers memory. Processors come in all different `shapes and sizes' -- there are many different types of architectures which are suited to a variety of different tasks.

Main memory (RAM -- Random Access Memory) -- used to store values during execution of a program. It can be written to and read from at any time.

Disc drive (hard or floppy) -- `permanently' stores files (programs and data). Hard discs are generally located inside the machine and come in a variety of different sizes and speeds. They do not, in fact, store files permanently -- they often go wrong and so must undergo a back-up at regular intervals. The floppy disc drive allows a user to make his or her own back up of important files and data. It is very important to keep back-ups.

Keyboard -- allows user to input information. Nowadays, most keyboards have more or less the same functionality.

VDU (Visual Display Unit) visually outputs data. There are numerous types of VDU differing in the resolution (dots per inch) and the number of colors that can be represented.

Printer -- allows a hard copy to be made. Again, there are many different types of printers available, for example, line printers, dot-matrix printers, bubble jet printers and laser printers. These also differ in their resolution and colour palette.

What is Hardware and Software?

A computer system is made up from hardware and software.

Hardware is the physical medium, for example: circuit boards processors keyboard

A piece of software is a computer program, for example: an operating system an editor
a compiler a Fortran 90 program The software allows the hardware to be used. Programs vary enormously in size and complexity.

Booting

Booting means restarting your computer. It is of two types:

Warm Boot

Restarting the computer without turning power off.

Cold boot

Restarting the computer when the power has been turned off then back on.

PRACTICAL 2

Aim: Navigating with Window Explorer, Working with Control Panel, To work at the command prompt,To understand the working of virus guards and anti virus software , Exploring the Internet.

Theory:

Windows operating System

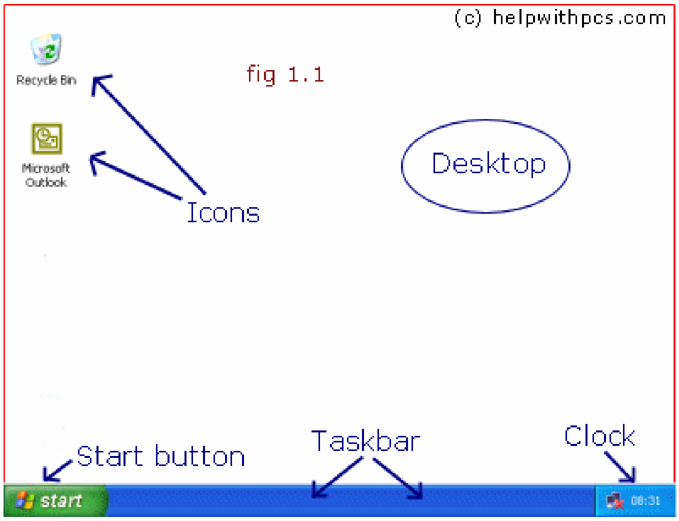
A family of operating systems for personal computers. Windows dominates the personal computer world, running, by some estimates, on 90% of all personal computers. The remaining 10% are mostly Macintosh computers.

Windows provides the following features Graphical user interface (GUI)

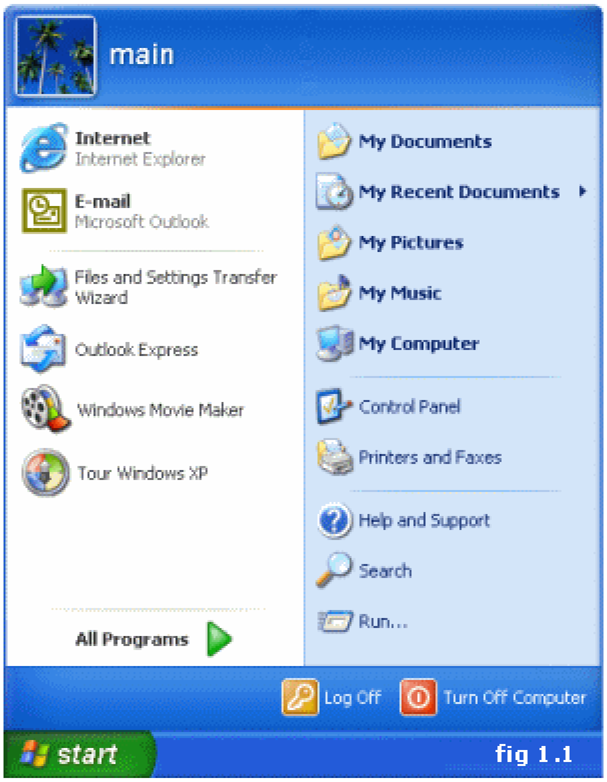
1) Virtual memory management
2) Multitasking
3) Support for many peripheral devices

Windows Desktop

Desktop refers to the main background area, we can customize your desktop by adding background pictures, changing the background color, changing the size of the icons, and more



An example of the Windows XP desktop, when we first switch on our machine and Windows XP loads.



Here is an overview of the different options:

Logoff - Log off the current user.

Turn Off Computer - shutdown, restart and standby.

All Programs - Access to installed programs.

Run - Used to manually start executable files (programs).

Search - Search your PC for pictures, music, documents, files and folders.

Help and Support - Opens the built-in Windows help including various help topics, windows updates and also allows you to request remote assistance from a friend/colleague via the internet or network.

***Printers and Faxes* -** Access to your printers and faxes folder, allows you to add and configure printers/faxes.

Control Panel - The control panel is used to configure various Windows XP settings, we will be looking at the Win XP control panel in detail later in the tutorial.

***My Computer* -** Gives you access to your computer's disk drives and files, we will be looking more closely at the My Computer function later in the tutorial.

***My Music* -** Links to a folder created by Windows XP which is used (by default) to store any music files on your hard drive.

***My Pictures* -** Links to a folder (again created by Windows XP) used (by default) to store any pictures/images on your hard drive.

My Recent Documents - This folder contains any recently viewed documents.

My Documents - Gives access to a folder created by Windows XP which is used (by default) to store any documents on your hard drive.

Tour Windows XP - Starts the built-in tour of Windows XP's features.

Windows Movie Maker - Opens Windows XP's movie editing software.

Outlook Express - Opens the Outlook Express mail program.

***Files and Settings Transfer Wizard* -** Allows you to import or export your files and settings from or to a different installation of Windows XP.

Searching Files in Windows:

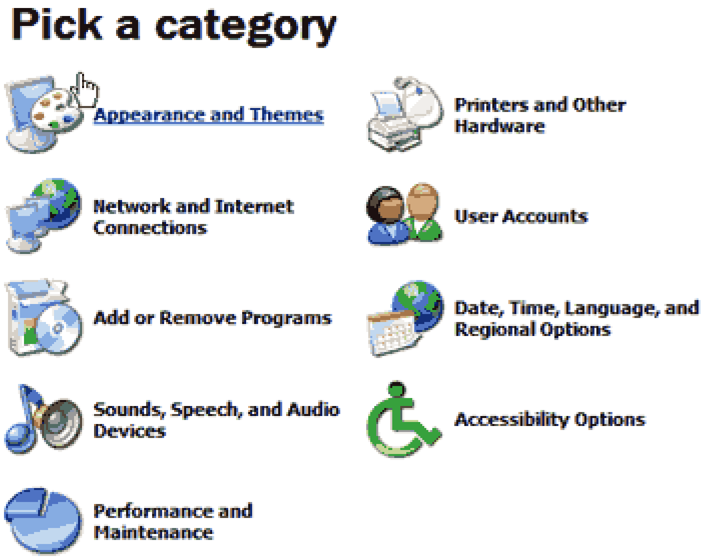
To search for a file or folder:

1. Click Start, and then click Search.
2. In the Search Companion dialog box, click All files and folders.
3. Type part or all of the name of the file or folder, or type a word or phrase that is in the file.
4. In the Look in box, click the drive or drives, folder, or network location that you want to search.

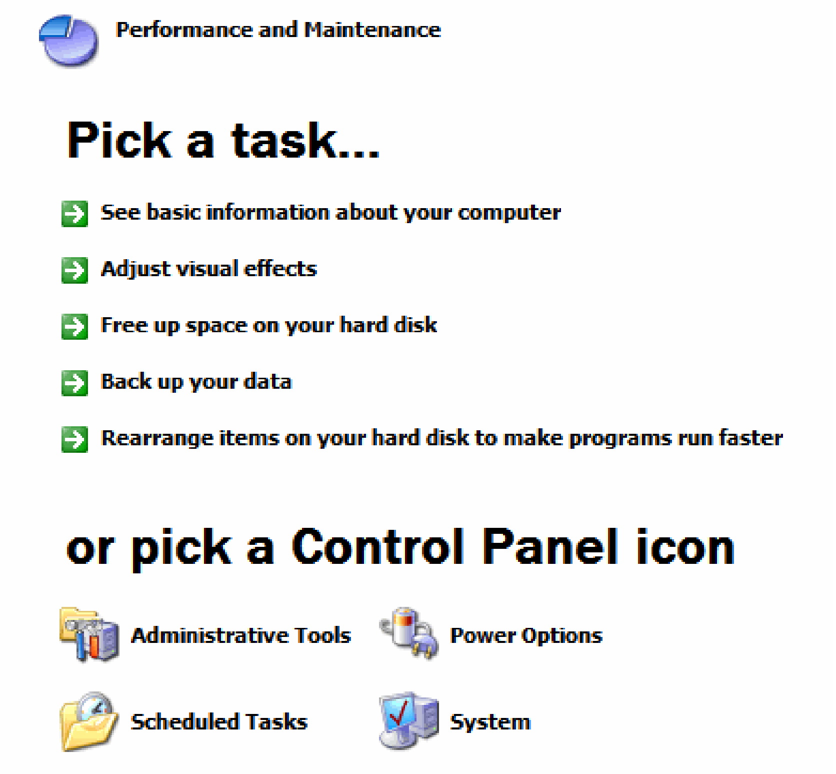
5. Choose one of the following options:
    a. Click When was it modified to look for files that were created or modi or between specific dates.
    b. Click What size is it to look for files that are specific size.
    c. Click More advanced options to specify additional search criteria.
6. Click Search.

CONTROL PANEL

The standard way to open Control Panel is through Start-Control Panel. There are two methods of displaying the contents. One is called the "Category View" and displays tasks by generalized categories as shown in the figure below.


Pick a category
Appearance and Themes
Printers and Other Hardware
Network and Internet Connections
User Accounts
Add or Remove Programs
Date, Time, Language, and Regional Options
Sounds, Speech, and Audio Devices
Accessibility Options
Performance and Maintenance

Choosing a category leads to another box with a further choice of tasks or icons for specific control panel applets.



Performance and Maintenance

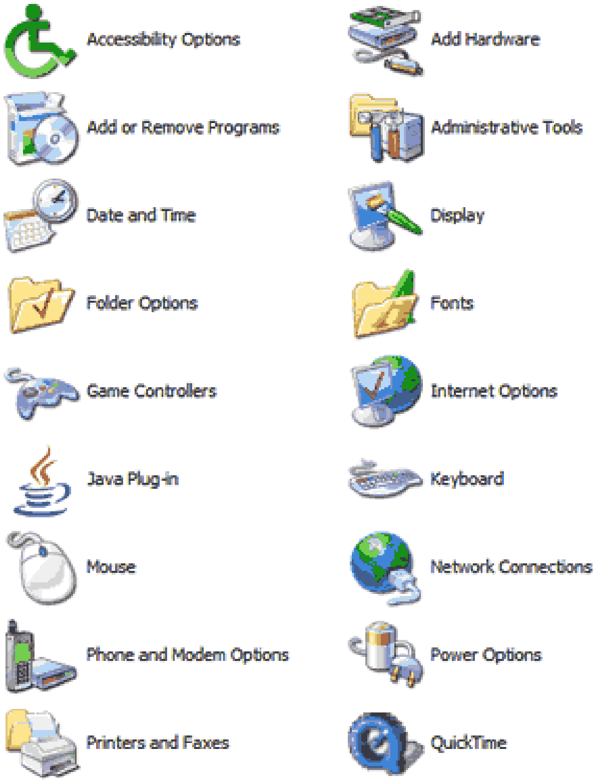
Pick a task...

- See basic information about your computer
- Adjust visual effects
- Free up space on your hard disk
- Back up your data
- Rearrange items on your hard disk to make programs run faster

or pick a Control Panel icon

Administrative Tools
Power Options
Scheduled Tasks
System

A second way of displaying Control Panel is called the "Classical View"



Accessibility Options
Add Hardware
Add or Remove Programs
Administrative Tools
Date and Time
Display
Folder Options
Fonts
Game Controllers
Internet Options
Java Plug-in
Keyboard
Mouse
Network Connections
Phone and Modem Options
Power Options
Printers and Faxes
QuickTime

COMMAND PROMPT

A **command-line interface** (**CLI**), also known as **command-line user interface**,**console user interface**,and **character user interface**(**CUI**), is a means of interacting with a computer program where the user (orclient) issues commands to the program in the form of successive lines of text (command lines).

The CLI was the primary means of interaction with most computer systems until the introduction of the video display terminal in the mid-1960s, and continued to be used throughout the 1970s and 1980s on Unix systems and personal computer systems including MS-DOS, CP/M and Apple DOS. The interface is usually implemented with a command line shell, which is a program that accepts commands as text input and converts commands to appropriate operating system functions.

Command-line interfaces to computer operating systems are less widely used by casual computer users, who favor graphical user interfaces. Command-line interfaces are often preferred by more advanced computer users, as they often provide a more concise and powerful means to control a program or operating system.

Programs with command-line interfaces are generally easier to automate via scripting.

Alternatives to the command line include, but are not limited to text user interface menus, keyboard shortcuts, and various other desktop metaphors centered on the pointer (usually controlled with a mouse).

Operating system command-line interfaces

Operating system (OS) command line interfaces are usually distinct programs supplied with the operating system.

A program that implements such a text interface is often called a command-line interpreter, command processor or shell. The term 'shell', often used to describe a command-line interpreter, can be in principle any program that constitutes the user-interface, including fully graphically oriented ones—for example, the default Windows GUI is created by a shell program named EXPLORER.EXE, as defined in the SHELL=EXPLORER.EXE line in the WIN.INI configuration file.

Examples of command-line interpreters include the various Unix shells (sh, ksh, csh, tcsh, bash, etc.), the historical CP/M CCP, and MS-DOS/IBM-DOS/DR-

DOS'sCOMMAND.COM, as well as the OS/2 and the Windows CMD.EXE programs, the latter groups being based heavily on DECs RSX and RSTS CLIs. Under most operating systems, it is possible to replace the default shell program with alternatives; examples include 4DOS for DOS, 4OS2 for OS/2, and 4NT or Take Command for Windows.

VIRUS RECOGNITION AND PROTECTION

A computer virus is a program designed to spread itself by first infecting Executable files or the system areas of hard and floppy disks and then making copies of itself. Viruses usually operate without the knowledge or desire of the computer user.

Viruses have the potential to infect any type of executable code, not just the files that are commonly called 'program files'. For example, some viruses infect executable code in the boot sector of floppy disks or in system areas of hard drives. Another type of virus, known as a 'macro' virus, can infect word processing and spreadsheet documents that use macros. And it's possible for HTML documents containing JavaScript or other types of executable code to spread viruses or other malicious code.

Since virus code must be executed to have any effect, files that the computer treats as pure data are safe. This includes graphics and sound files such as .gif, .jpg, .mp3, .wav, etc., as well as plain text in .txt files. For example, just viewing picture files won't infect your computer with a virus. The virus code has to be in a form, such as an .exe program file or a Word .doc file, that the computer will actually try to execute.

When you execute program code that's infected by a virus, the virus code will also run and try to infect other programs, either on the same computer or on other computers connected to it over a network. And the newly infected programs will try to infect yet more programs.

When you share a copy of an infected file with other computer users, running the file may also infect their computers; and files from those computers may spread the infection to yet more computers.

If your computer is infected with a boot sector virus, the virus tries to write copies of itself to the system areas of floppy disks and hard disks. Then the infected floppy disks may infect other computers that boot from them, and the virus copy on the hard disk will try to infect still more floppies.

Some viruses, known as 'multipartite' viruses, can spread both by infecting files and by infecting the boot areas of floppy disks.

Viruses are software programs, and they can do the same things as any other programs running on a computer. The actual effect of any particular virus depends on how it was programmed by the person who wrote the virus.

Some viruses are deliberately designed to damage files or otherwise interfere with your computer's operation, while others don't do anything but try to spread themselves around. But even the ones that just spread themselves are harmful, since they damage files and may cause other problems in the process of spreading.

Note that viruses can't do any damage to hardware: they won't melt down your CPU, burn out your hard drive, cause your monitor to explode, etc. Warnings about viruses that will physically destroy your computer are usually hoaxes, not legitimate virus warnings.

A type of program that is often confused with viruses is a 'Trojan horse' program. This is not a virus, but simply a program (often harmful) that pretends to be something else.

For example, you might download what you think is a new game; but when you run it, it deletes files on your hard drive. Or the third time you start the game, the program E-mails your saved passwords to another person.

Note: simply downloading a file to your computer won't activate a virus or Trojan horse; you have to execute the code in the file to trigger it. This could mean running a program file, or opening a Word/Excel document in a program (such as Word or Excel) that can execute any macros in the document.

You can't get a virus just by reading a plain-text E-mail message or Usenet post. What you have to watch out for are encoded messages containing embedded executable code (i.e., JavaScript in an HTML message) or messages that include an executable file attachment (i.e., an encoded program file or a Word document containing macros).

In order to activate a virus or Trojan horse program, your computer has to execute some type of code. This could be a program attached to an E-mail, a Word document you downloaded from the Internet, or something received on a floppy disk. There's no special hazard in files attached to Usenet posts or E-mail messages: they're no more dangerous than any other file.

Treat any file attachments that might contain executable code as carefully as you would any other new files: save the attachment to disk and then check it with an up-to-date virus scanner before opening the file.

If your E-mail or news software has the ability to automatically execute JavaScript, Word macros, or other executable code contained in or attached to a message, I strongly recommend that you disable this feature.

My personal feeling is that if an executable file shows up unexpectedly attached to an E-mail, you should delete it unless you can positively verify what it is, who it came from, and why it was sent to you.

The recent outbreak of the Melissa virus was a vivid demonstration of the need to be extremely careful when you receive E-mail with attached files or documents. Just because an E-mail appears to come from someone you trust, this does NOT mean the file is safe or that the supposed sender had anything to do with it.

Some general tips on avoiding virus infections:

1. Install anti-virus software from a well-known, reputable company, UPDATE it regularly, and USE it regularly.

New viruses come out every single day; an a-v program that hasn't been updated for several months will not provide much protection against current viruses.

2. In addition to scanning for viruses on a regular basis, install an 'on access' scanner (included in most good a-v software packages) and configure it to start automatically each time you boot your system. This will protect your system by checking for viruses each time your computer accesses an executable file.

3. Virus scan any new programs or other files that may contain executable code before you run or open them, no matter where they come from. There have been cases of commercially distributed floppy disks and CD-ROMs spreading virus infections.

4. Anti-virus programs aren't very good at detecting Trojan horse programs, so be extremely careful about opening binary files and Word/Excel documents from unknown or 'dubious' sources. This includes posts in binary newsgroups, downloads from web/ftp sites that aren't well-known or don't have a good reputation, and executable files unexpectedly received as attachments to E-mail or during an on-line chat session.

5. If your E-mail or news software has the ability to automatically execute JavaScript, Word macros, or other executable code contained in or attached to a message, I strongly recommend that you disable this feature.

6. Be _extremely_ careful about accepting programs or other files during on-line chat sessions: this seems to be one of the more common means that people wind up with virus or Trojan horse problems. And if any other family members (especially younger ones) use the computer, make sure they know not to accept any files while using chat.

7. Do regular backups. Some viruses and Trojan horse programs will erase or corrupt files on your hard drive, and a recent backup may be the only way to recover your data.

Ideally, you should back up your entire system on a regular basis. If this isn't practical, at least backup files that you can't afford to lose or that would be difficult to replace: documents, bookmark files, address books, important E-mail, etc.

Dealing with virus infections:

First, keep in mind "Nick's First Law of Computer Virus Complaints":

"Just because your computer is acting strangely or one of your programs doesn't work right, this does NOT mean that your computer has a virus."

1. If you haven't used a good, up-to-date anti-virus program on your computer, do that first. Many problems blamed on viruses are actually caused by software configuration errors or other problems that have nothing to do with a virus.

2. If you do get infected by a virus, follow the directions in your anti-virus program for cleaning it. If you have backup copies of the infected files, use those to restore the files. Check the files you restore to make sure your backups weren't infected.

3. For assistance, check the web site and support services for your anti-virus software.

Most personal computers are now connected to the Internet and to local area networks, facilitating the spread of malicious code. Today's viruses may also take advantage of network services such as the World Wide Web, e-mail, Instant Messaging and file sharing systems to spread, blurring the line between viruses and worms. Furthermore, some sources use an alternative terminology in which a virus is any form of self-replicating malware.

Replication strategies

In order to replicate itself, a virus must be permitted to execute code and write to memory. For this reason, many viruses attach themselves to executable files that may be part of legitimate programs. If a user tries to start an infected program, the virus' code may be executed first. Viruses can be divided into two types, on the basis of their behavior when they are executed. Nonresident viruses immediately search for other hosts that can be infected, infect these targets, and finally transfer control to the application program they infected. Resident viruses do not search for hosts when they are started. Instead, a resident virus loads itself into memory on execution and transfers control to the host program. The virus stays active in the background and infects new hosts when those files are accessed by other programs or the operating system itself.

Nonresident viruses

Nonresident viruses can be thought of as consisting of a finder module and a replication module. The finder module is responsible for finding new files to infect. For each new executable file the finder module encounters, it calls the replication module to infect that file.

Resident viruses

Resident viruses contain a replication module that is similar to the one that is employed by nonresident viruses. However, this module is not called by a finder module. Instead, the virus loads the replication module into memory when it is executed and ensures that this module is executed each time the operating system is called to perform a certain operation. For example, the replication module can be called each time the operating system executes a file. In this case, the virus infects every suitable program that is executed on the computer

Methods to avoid detection

In order to avoid detection by users, some viruses employ different kinds of deception. Some old viruses, especially on the MS-DOS platform, make sure that the "last modified" date of a host file stays the same when the file is infected by the virus. This approach does not fool anti-virus software, however, especially that which maintains and dates Cyclic redundancy check on file changes.

Some viruses can infect files without increasing their sizes or damaging the files. They accomplish this by overwriting unused areas of executable files. These are called cavity viruses. For example the CIH virus, or Chernobyl Virus, infects Portable Executable files. Because those files had many empty gaps, the virus, which was 1 KB in length, did not add to the size of the file.

Some viruses try to avoid detection by killing the tasks associated with antivirus software before it can detect them.
As computers and operating systems grow larger and more complex, old hiding techniques need to be updated or replaced. Defending a computer against viruses may demand that a file system migrate towards detailed and explicit permission for every kind of file access.

Anti-virus software and other preventive measures

Many users install anti-virus software that can detect and eliminate known viruses after the computer downloads or runs the executable. There are two common methods that an anti-virus software application uses to detect viruses. The first, and by far the most common method of virus detection is using a list of virus signature definitions. This works by examining the content of the computer's memory (its RAM, and boot sectors) and the files stored on fixed or removable drives (hard drives, floppy drives), and comparing those files against a database of known virus "signatures". The disadvantage of this detection method is that users are only protected from viruses that pre-date their last virus definition update. The second method is to use a heuristic algorithm to find viruses based on common behaviors. This method has the ability to detect viruses that anti-virus security firms have yet to create a signature for.

Some anti-virus programs are able to scan opened files in addition to sent and received e-mails 'on the fly' in a similar manner. This practice is known as "on-access scanning." Anti-virus software does not change the underlying capability of host software to transmit viruses. Users must update their software regularly to patch security holes. Anti-virus software also needs to be regularly updated in order to prevent the latest threats.

One may also prevent the damage done by viruses by making regular backups of data (and the Operating Systems) on different media, that are either kept unconnected to the system (most of the time), read-only or not accessible for other reasons, such as using different file systems. This way, if data is lost through a virus, one can start again using the backup (which should preferably be recent). If a backup session on optical media like CD and DVD is closed, it becomes read-only and can no longer be affected by a virus. Likewise, an Operating System on a bootable can be used to start the computer if the installed Operating Systems become unusable. Another method is to use different Operating Systems on different file systems. A virus is not likely to affect both. Data backups can also be put on different file systems. For example, Linux requires specific software to write to NTFS partitions, so if one does not install such software and uses a separate installation of MS Windows to make the backups on an NTFS partition, the backup should remain safe from any Linux viruses. Likewise, MS Windows can not read file systems like ext3, so if one normally uses MS Windows, the backups can be made on an ext3 partition using a Linux installation.

EXPLORING INTERNET

INTERNET

The Internet is a worldwide, publicly accessible series of interconnected computer networks that transmit data by packet switching using the standard Internet Protocol (IP). It is a "network of networks" that consists of millions of smaller domestic, academic, business, and government networks, which together carry various information and services, such as electronic mail, online chat, file transfer, and the interlinked web pages and other resources of the World Wide Web (WWW).

WORLD WIDE WEB

The World Wide Web (commonly shortened to the Web) is a system of interlinked hypertext documents accessed via the Internet. With a Web browser, a user views Web pages that may contain text, images, videos, and other multimedia and navigates between them using hyperlinks. The World Wide Web was created in 1989 by Sir Tim Berners-Lee, working at CERN in Geneva, Switzerland. Since then, Berners-Lee has played an active role in guiding the development of Web standards .

WEB BROWSER

A web browser is a software application which enables a user to display and interact with text, images, videos, music and other information typically located on a Web page at a website on the World Wide Web or a local area network. Text and images on a Web page can contain hyperlinks to other Web pages at the same or different website. Web browsers allow a user to quickly and easily access information provided on many Web pages at many websites by traversing these links. Web browsers format HTML information for display, so the appearance of a Web page may differ between browsers .

Some of the Web browsers available for personal computers include Internet Explorer, Mozilla Firefox, Safari, and Opera in order of descending popularity (in November 2007).[3] Web browsers are the most commonly used type of HTTP user agent. Although browsers are typically used to access the World Wide Web, they can also be used to access information provided by Web servers in private networks or content in file systems.

Protocols and standards

Web browsers communicate with Web servers primarily using HTTP (hypertext transfer protocol) to fetch webpages. HTTP allows Web browsers to submit information to Web servers as well as fetch Web pages from them. The most commonly used HTTP is HTTP/1.1, which is fully defined in RFC 2616. HTTP/1.1 has its own required standards that Internet Explorer does not fully support, but most other current-generation Web browsers do.

Pages are located by means of a URL (uniform resource locator, RFC 1738), which is treated as an address, beginning with http: for HTTP access. Many browsers also support a variety of other URL types and their corresponding protocols, such as gopher: for Gopher (a hierarchical hyperlinking protocol), ftp: for FTP (file transfer protocol), rtsp: for RTSP (real-time streaming protocol), and https: for HTTPS (an SSL encrypted version of HTTP). The file format for a Web page is usually HTML (hyper-text markup language) and is identified in the HTTP protocol using a MIME content type. Most browsers natively support a variety of formats in addition to HTML, such as the JPEG, PNG and GIF image formats, and can be extended to support more through the use of plugins. The combination of HTTP content type and URL protocol specification allows Web page designers to embed images, animations, video, sound, and streaming media into a Web page, or to make them accessible through the Web page.

Early Web browsers supported only a very simple version of HTML. The rapid development of proprietary Web browsers led to the development of non-standard dialects of HTML, leading to problems with Web interoperability. Modern Web browsers support a combination of standards- and defacto-based HTML and XHTML, which should display in the same way across all browsers. No browser fully supports HTML 4.01, XHTML 1.x or CSS 2.1 yet. Currently many sites are designed using WYSIWYG HTML generation programs such as Macromedia Dreamweaver or Microsoft FrontPage. These often generate non-standard HTML by default, hindering the work of the W3C in developing standards, specifically with XHTML and CSS

Some of the more popular browsers include additional components to support Usenet news, IRC (Internet relay chat), and e-mail. Protocols supported may include NNTP (network news transfer protocol), SMTP (simple mail transfer protocol), IMAP (Internet message access protocol), and POP (post office protocol). These browsers are often referred to as Internet suites or application suites rather than merely Web browsers.

History

One important person in the early design and evolution of the browser was Neil Larson who in 1977 created a TRS-80 program that displayed outlines a level at a time with hypertext jumps between levels In 1984, expanding on ideas from futurist Ted Nelson, his commercial DOS Maxthink outline program added angle bracket hypertext jumps (adopted by later web browsers) to and from ASCII, batch, and other Maxthink files up to 32 levels deep.[citation needed] In 1986 he released his DOS Houdini network browser program that supported 2500 topics cross-connected with 7500 links in each file along with hypertext links among unlimited numbers of external ASCII, batch, and other Houdini files

In 1987, these capabilites were included in his then popular shareware DOS file browser programs HyperRez (memory resident) and PC Hypertext (which also added jumps to programs, editors, graphic files containing hot spots jumps, and cross-linked theraurus/glossary files). These programs introduced many to the browser concept and 20 years later, Google still lists 3,000,000 references to PC Hypertext

PRACTICAL 3

AIM: To study application software Open Office.org-Writer, Calc and Impress.

Theory:

A) Writer

Open A New Text Document

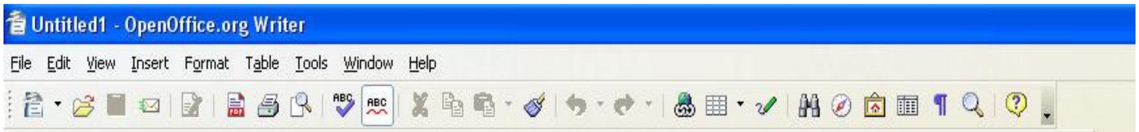
IF on the desktop, click start> **All Programs > OpenOffice.org 2.3 > OpenOffice.org Writer**.

IF you are in **OpenOffice.org Writer**, click **File > New > Text Document**.

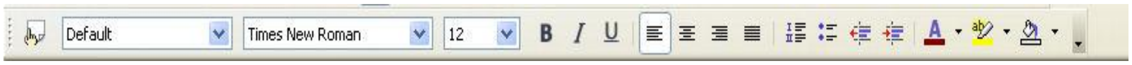
In either case, A text document appears on your screen. (The name of the text document Untitled1 - OpenOffice.org Writer appears at the top of the screen.

The next line down that starts with the word **File** is called the **Menu Bar**. (Clicking on a word in the Menu Bar shows a list of things that you can choose to do in Writer. This list is called a **Menu.**)

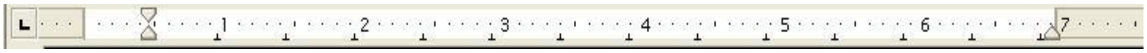
The third line down is the **Standard Toolbar**. (Clicking on a small picture will make available a specific function that can be done in Writer.)


Untitled1 - OpenOffice.org Writer

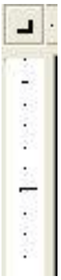
The fourth line down is the **Formatting Toolbar**. (Additional specific functions that can be done in Writer are on this line.)


Default
Times New Roman
12

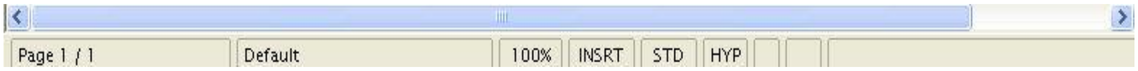
7. The fifth line down is the **Ruler** which appears on the top and the left side of the page. (Click "View". If there is a check mark by **Ruler**, a ruler will appear at the top and the left side of your page on the screen.) The **Horizontal Ruler** is below.



8. The numbers on the **Ruler** on the left side of the page indicate where the text is located on the page from the top of the page. The numbers at the top of the page indicate the location of the text from the left or right sides of the page.)



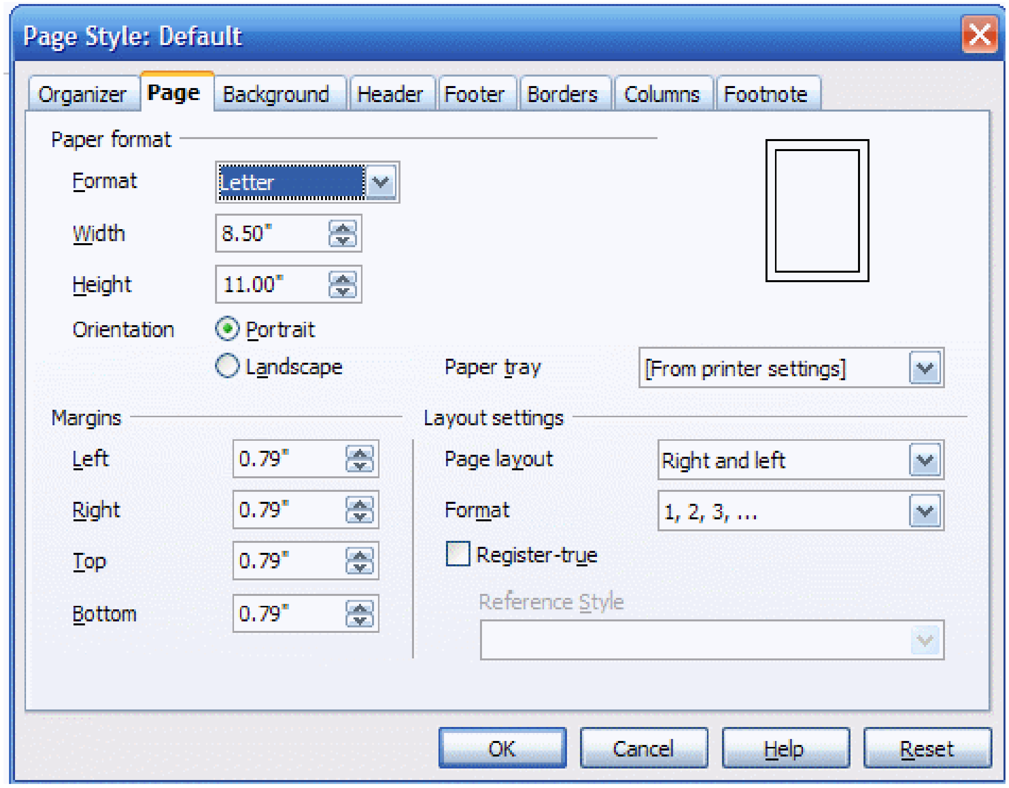
9. Right-click on the **Ruler** to see the choices for the settings which are **Millimeter, Centimeter, Inch, Point,** and **Pica**.
10. The **Horizontal Scroll bar** (by clicking and holding the pointer, you can navigate a document left and right) is located at the bottom of the page above the **Status Bar** which is the line at the bottom of the screen that starts with **Page 1/1.** (This line gives you information about the document you are presently using.)



11. The **Vertical Scroll Bar** (by clicking and holding the pointer, you can navigate up and down) is located on the right side of your screen.

Paper Size

1. Click **Format** > **Page** > **Page** tab. (The “Page Style: Default” window appears)



Page Style: Default

Paper format

Format: Letter

Width: 8.50"

Height: 11.00"

Orientation: ● Portrait ○ Landscape

Paper tray: [From printer settings]

Margins

Left: 0.79"

Right: 0.79"

Top: 0.79"

Bottom: 0.79"

Layout settings

Page layout: Right and left

Format: 1, 2, 3, ...

□ Register-true

Reference Style

2. For this tutorial we will use a paper size of 8 ½ by 11 inches which is a common size in the USA. Under “Paper format”, in the “Format” pull-down menu, select **Letter** if it is not already selected. To use other paper sizes, open the “Format” pull-down menu and select a size of your choice.

3. Words, numbers, and graphics can be placed anywhere on a sheet of paper except for a small border at the four outer edges of the paper. The word **margin** is used to describe the boundary lines where the borders end and the words begin. All future added words, numbers, and graphics will stay within the lines that indicate the margins. The margin lines appear on the screen and will not appear on the printed page.

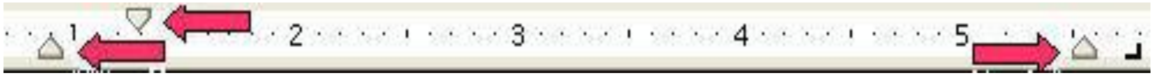
Change Margins

We will change the margins so there is 7 inches across the page instead of 6 inches. To do this both the left margin and the right margin is reduced from 1.25” to .75”.

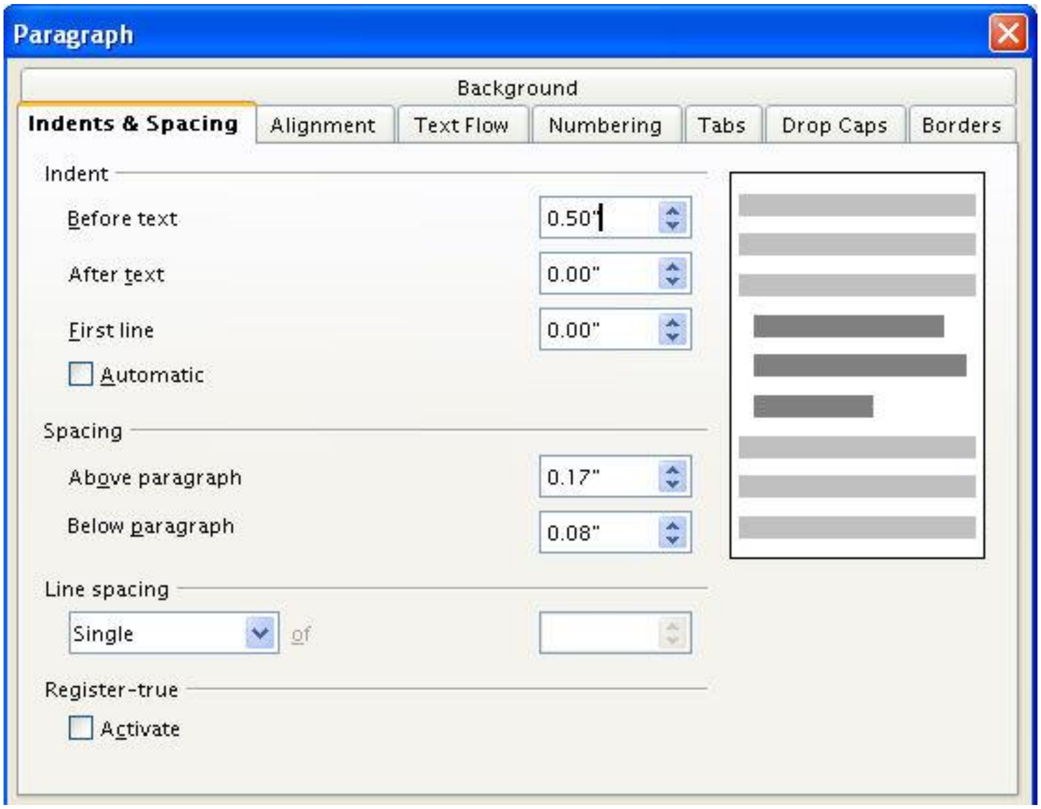
1. In the “Page Style: Default” window, Click the **Page** tab if it is not already selected. Under “Margins”, In the **Left** dialog box, change whatever number is there to **0.75”**.
2. One way to get **0.75”** is to click repeatedly on the ⬍until **.75** appears in the **Left** dialog box (The up-facing triangle increases the number and the down-facing triangle decreases the number).
3. Another way to get **0.75”** is to type **.75** (Click on the number that is there then type 75 then delete the original number by pressing the Delete key)
4. In the **Right** dialog box, change the **1.25”** to **.75”**, (This is done the same way that you changed the “Left” dialog box)
5. Click the **OK** button. (The “Page Style: Default” window closes.)

Use The Ruler To Change The Indentation Of Selected Paragraphs

Indents are adjusted with the three small triangles on the horizontal ruler or use the “Paragraph” window > “Indents & Spacing” tab by double-clicking anywhere on the horizontal ruler.



1. To change the **left** or the **right** paragraph indent, highlight the **paragraph(s)** that you want to change the **indent**, drag the **bottom left** or the **bottom right triangle** on the horizontal ruler to a new location.
2. To change the **first line indent** of a selected paragraph, drag the **top left triangle** on the **Horizontal Ruler** to a new location.
3. You can adjust the **indents** by double-clicking anywhere on the **Horizontal Rule**r and adjust the **indents** in the **Paragraph** dialog. (The “Paragraph” window appears.)



Indent

Before text 0.50"

After text 0.00"

First line 0.00"

□ Automatic

Spacing

Above paragraph 0.17"

Below paragraph 0.08"

Line spacing

Single of

Register-true

□ Activate

Icons, Click, Right-click, Insert

1. The small pictures on the **Standard Toolbar** and some of the other toolbars are called **<u>icons</u>**. Click on each icon. (**<u>Click</u>** means to press and release the button on the left side of the mouse with your index (pointer) finger. Click only one time unless instructed to "double- click".)
2. Find the **Gallery** icon, . Click on the **Gallery** icon. (The **Gallery** window appears on the screen. The background changes color when the icon is in use. Some users may see a white background and some may see a blue background.)
3. Click on the **Gallery** icon again to close the window. (There is no background color on the "Gallery" icon.)
4. **Right-click** anywhere on the printed or white page area of this tutorial. (**<u>Right-click</u>** means to press and release the button on the right side of the mouse with your middle finger. Click only one time unless instructed to double-click. A menu appears on the page. To close the window, click anywhere on the page other than on the menu.)
5. On the **Menu Bar**, click on **<u>I</u>nsert**. (The "Insert" menu appears. This is a list of the various things you can choose to do in Writer.)

Turn Off The Word Completion Feature

1. Click **Tools > AutoCorrect > Word Completion** tab. (The “AutoCorrect” window appears.)
2. Remove the check-mark in the box, if it is there, before the words **Enable word completion**. Click on the **OK** button.

Close And Open Files

1. On the **Menu Bar**, click on the word **File**. (A menu opens that has a list of words.)
2. Click on the words, **Save As**. (The “Save As” window appears.)
3. Click **My Documents > File name:** (If the words, "My Documents", are not already in the "Save in:" dialog box, click the ▼ at the right end of the “Save in:' dialog box. A menu will appear; find “My Documents"” on this menu and click it. "My Documents" appears in the "Save in:" dialog box. If the cursor is not flashing in the “File name:” dialog box at the bottom of the “Save As:” window, move the I-beam pointer into the box and click. The cursor is flashing in the “File name:” dialog box.)
4. In the **File name:** dialog box, type the words, **My Lesson.** In the **Save as type:** pull-down menu box, click **OpenDocument Text (.odt)** if it is not already selected. Click on the **Save** button located to the right of the **File name:** dialog box. (The “My Lesson” file has been saved on your computer in the folder, "My Documents", and can be opened later. The top line on the screen has changed to My Lesson.odt - OpenOffice.org Writer. This indicates the file “My Lesson” is still open on your computer.)
5. Click the word, **File,** again. Click the word, **Close**. (The “My Lesson” file is closed and is not seen on the screen.)
6. Click **File > Open**. (The “Open” window appears. If "My Documents" is not already in the **Look in:** dialog box at the top of the "Open" window, in the list of directories below the **Look in:** box, click "My Documents". The words, "My Documents", appear in the "Look in:" dialog box.)
7. Click on **My Lesson** in the list below the **Look in:** dialog box. (The words, “My Lesson”, appear in the “File name:” dialog box.)
8. Click the **Open** button which is to the right of the **File name:** dialog box. (The top line on the screen changes to My Lesson.odt - OpenOffice.org Writer and the first page of the file, “My Lesson”, appears on the screen below the ruler.)

Undo Entries

Entries can be undone (can be deleted) in the reverse order in which the entries were made. Word(s), graphic(s), and paragraph(s) can be deleted and "Undo" can be used to put them back in your document. Type the three paragraphs below.

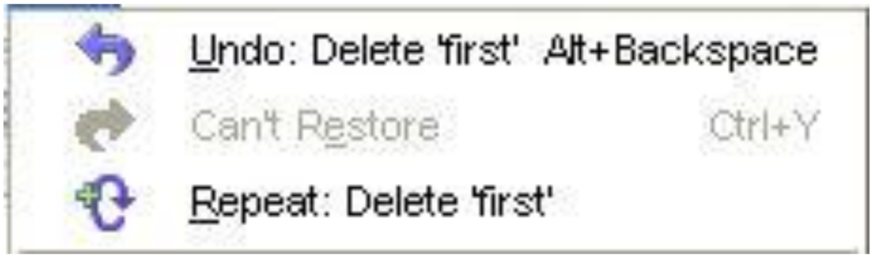
This is the first paragraph.

This is the second paragraph.

This is the third paragraph.

Delete Word(s)

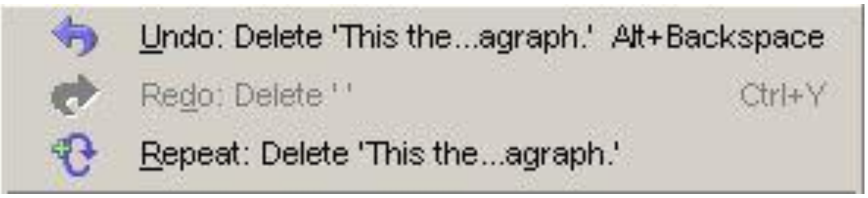
1. In the first paragraph, (**This is the first paragraph.**), highlight the word, **first.** Press the **delete** key. (The word, “first”, is deleted.)
2. Click **Edit**. (You will see the menu below.)



3. Click on **Undo: Delete 'first'**. (The word, "first", that was deleted is placed back in the document and the menu will close.)

Delete One Paragraph

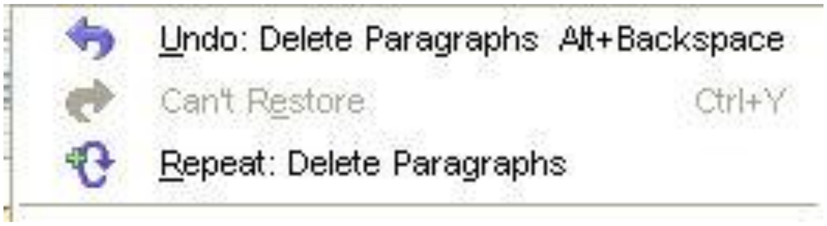
1. Highlight the first paragraph, **This is the first paragraph.** Press the **Delete** key to delete the paragraph. Click **Edit.** (When you delete one paragraph and you click **Edit**, you will see the menu below,


Undo: Delete 'This the...agraph.' Alt+Backspace
Redo: Delete ' ' Ctrl+Y
Repeat: Delete 'This the...agraph.'

2. Click **Undo: Delete 'This is ...agraph.'** (The paragraph, "This is the first paragraph" is placed back in the document. The length of the sentences in the paragraph determines how much of the paragraph is shown after the "Undo: Delete".)

Delete More Than One Paragraph

1. Highlight the three paragraphs (This is the first paragraph. This is the second paragraph. This is the third paragraph.) at the beginning of the section, **Undo Entries**. Press the **delete** key to delete the paragraphs. (The three paragraphs are deleted.)
2. Click **Edit**. (You will see the menu below.)

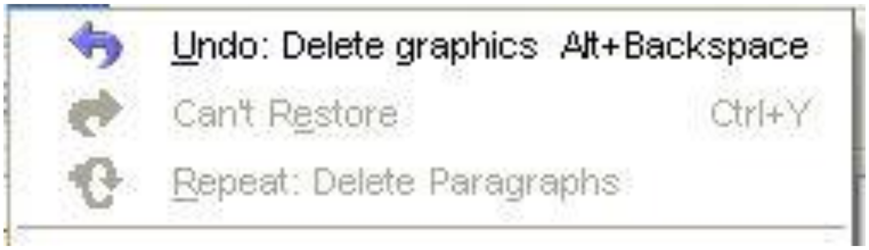

Undo: Delete Paragraphs Alt+Backspace
Can't Restore Ctrl+Y
Repeat: Delete Paragraphs

3. Click on the **Undo: Delete 'Paragraphs'**. (The deleted paragraphs are placed back in the document.)

Delete Graphic(s)

1. Click on the image below. (The green handles will appear.)

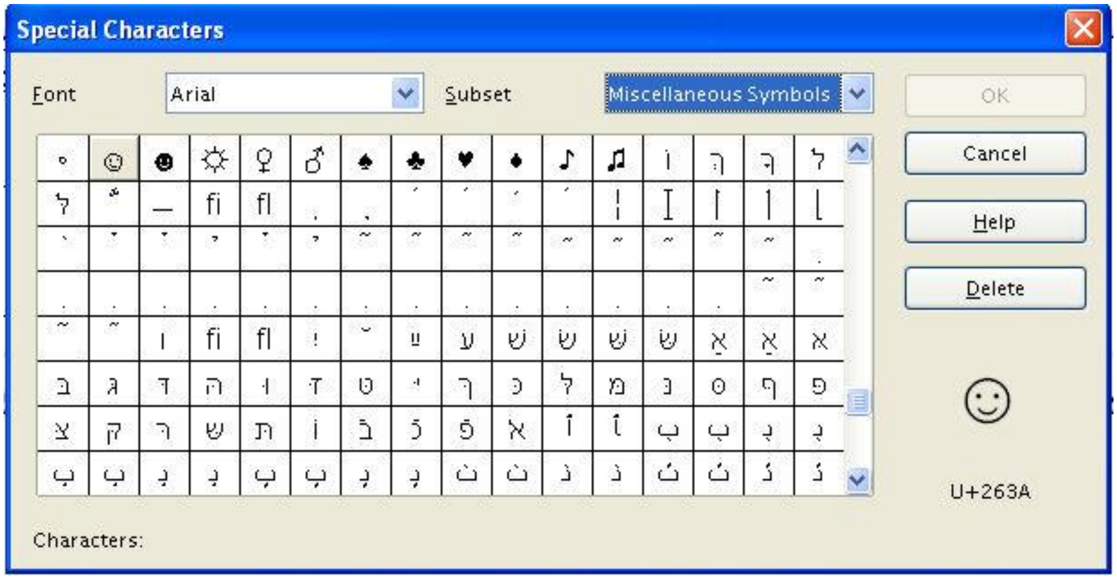
2. Press the **Delete** key. (The graphic is no longer on your document.)
3. Click **Edit**. (You will see the menu below.)


Undo: Delete graphics Alt+Backspace
Can't Restore Ctrl+Y
Repeat: Delete Paragraphs

4. Click **Undo: Delete graphics Alt+Backspace.** (The graphic reappears on the document.)

Insert Special Character

1. Place the pointer on the page and click where the symbol is to appear.
2. Click **Insert** > **Special Character**. (The "Special Characters" window appears. You will see two dialog boxes. The one on the left is the "Font" menu and the one on the right is the “Subset” menu.)



Special Characters

Font: Arial

Subset: Miscellaneous Symbols

U+263A

Characters:

3. In the **Subset** menu**,** click on the ▾ until you see **Miscellaneous Symbols.** Click **Miscellaneous Symbols.**
4. Scroll until you see the "☺" symbol. Click on the **"☺"** .
5. Click **OK.**
6. The ☺symbol appears on the page.

Getting Help

1. On the **Menu Bar**, click on the word **Help**. (A menu appears)
2. Click on the words, **What's this?** (The mouse pointer becomes ↖?).
3. On the **Standard Toolbar**, move the pointer onto a picture (icon). (A brief description of what the picture does appears on the screen)
4. On the **Standard Toolbar** and the **Formatting Toolbar**, move the pointer onto the **icons** to see what is there.
5. To get rid of the question mark, click anywhere on the screen below the **Standard Toolbar** and the **Formatting Toolbar**.

B) CALC

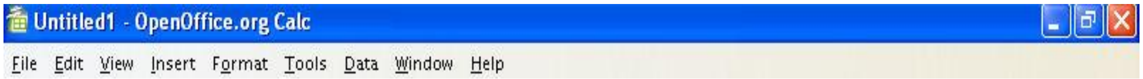
Open A Spreadsheet

1. **IF** on the desktop, click start> **All Programs > OpenOffice.org 2.3 > OpenOffice.org Calc**.
2. **IF** you are in a OpenOffice program, click **File > New > Spreadsheet**.
3. In either case a spreadsheet called Untitled1 - OpenOffice.org Calc appears on our screen.

The Calc Toolbars

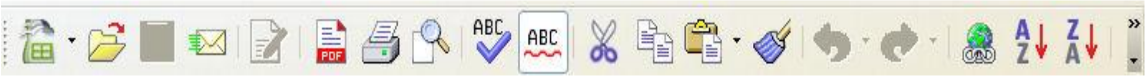
The following four Calc Toolbars appear at the top of all Calc screens

Main Menu Toolbar



The first toolbar is the **Main Menu** toolbar that gives you access to many of the basic commands used in Calc.

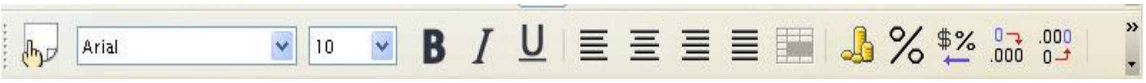
Function Toolbar



The second toolbar down is the **Function Toolbar**. The **Function Toolbar** contains **icons** (pictures) to provide quick access to commands like **New**, **Open**, **Print**, **Copy**, **Paste**, etc. When you place your mouse cursor over any of the elements of a toolbar, the name of the element appears on your screen.

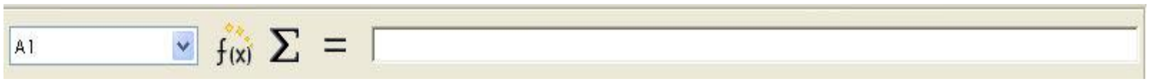
Move your cursor over the icon . (The word "New" appears. Clicking on opens a new spreadsheet.)

Formatting Toolbar



The third toolbar down is the **Formatting Toolbar**. The **Formatting Toolbar** has **icons** plus drop-down menus that allow you to select a **font, font color, alignments, number formats, border options and background colors.**

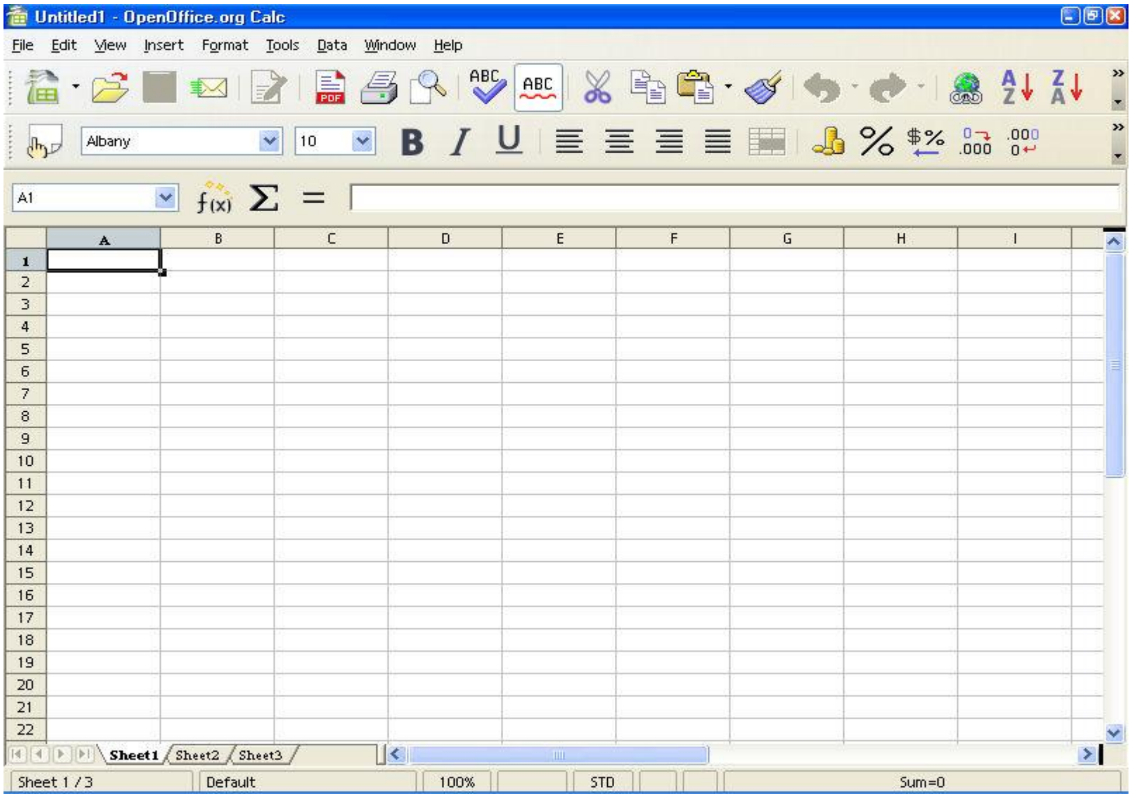
Formula Toolbar



The fourth toolbar down is the **Formula Toolbar**. The **Formula Toolbar** contains the **Name Box** drop-down menu and a long white box called the **Input Line**.

Note: If your Toolbars look different, it is because these toolbars are in 800x600 screen resolution and the last eight icons are not shown but are available by clicking on the » on the far right of the toolbar.

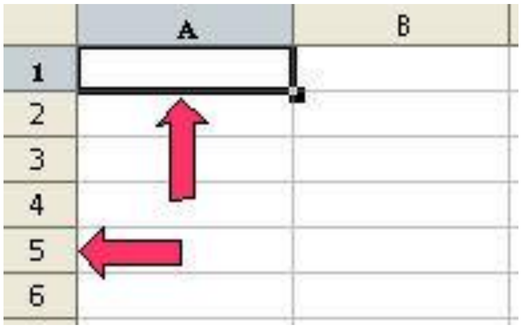
The Spreadsheet Itself



The rest of the window contains the **spreadsheet**. The spreadsheet is divided into rows that have a number at the left of each row and divided into columns with letters at the top of each column.

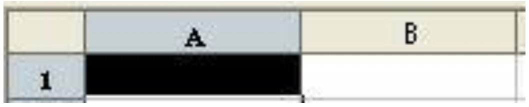
Cells

A **cell** is the fundamental element of a worksheet. This is where things are added and where things are seen. A **cell address** in a spreadsheet identifies the location of the cell in the spreadsheet. A **cell address** is a combination of the column letter and the row number of a cell, such as **A2** or **B16**.etc. When identifying a cell by its address, the column letter is always listed first followed by the row number. The cell address of the example below is **A5**.

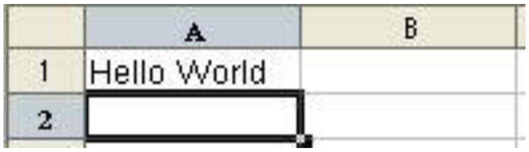


Enter Data

1. Click on the **A1** cell (The cell at the very top left of the spreadsheet).
2. Notice the heavy black border around the **A1** cell. The heavy black border indicates that **A1** is the **active cell**. (**A** is at the top of the first column and **1** is on the first row. Both are highlighted. The highlighting also indicates that **A1** is the **active cell**.)



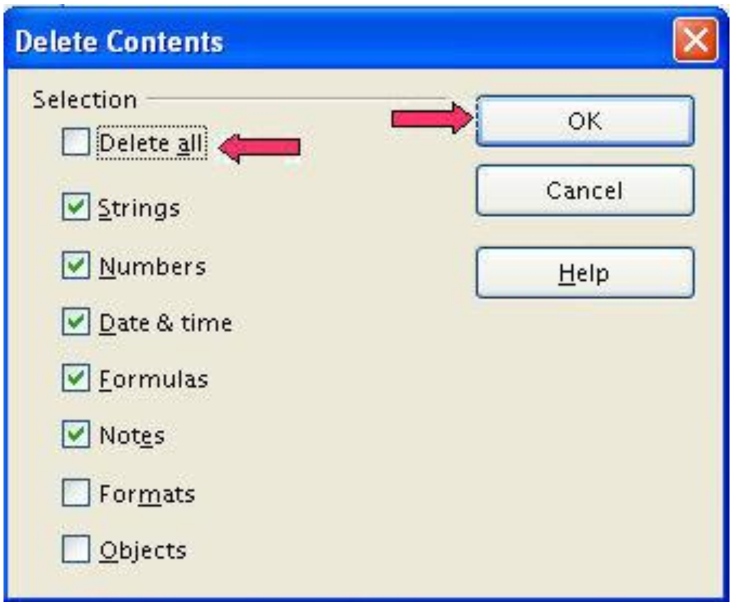
3. Type **Hello World** and press **Enter**.
4. The active cell is now **A2**. (The words "Hello World" are in A1.)



5. When you type something in a cell and press **Enter**, what you typed is seen in that cell and the cell below becomes the next active cell.

Delete Data

6. Click on **A1** again.
7. Press the **Delete** key. (The "Delete Contents" window appears.)



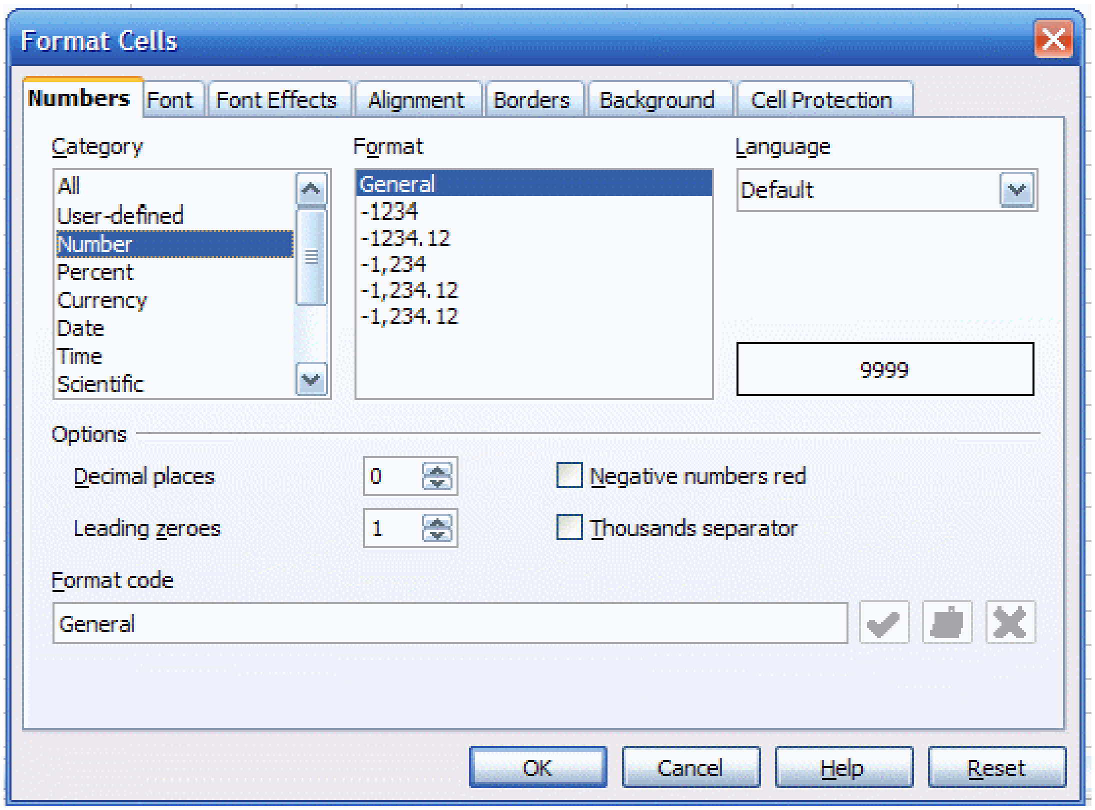
☐ Delete all
☑ Strings
☑ Numbers
☑ Date & time
☑ Formulas
☑ Notes
☐ Formats
☐ Objects

3. Check the **Delete all** box then click **OK.** ("Hello World" is deleted from A1)

Format Data

Separate Thousands, Two Decimal Places, Red Negtive Numbers

1. Click cell **A1**. > Type the number **-9999.129** > Press **Enter**. (The cursor moves to cell A2)
2. **Right-click** cell A1. (A small menu appears) > click **Format cells**. (The "Format cells" window appears)



Category
All
User-defined
Number
Percent
Currency
Date
Time
Scientific

Format
General
-1234
-1234.12
-1,234
-1,234.12
-1,234.12

Language: Default

9999

Options

Decimal places: 0

Leading zeroes: 1

□ Negative numbers red

□ Thousands separator

Format code: General

1. Click **Numbers** tab. Under “Category”, click on **Number**. Under “Format”, click on **-1,234.12**. Click In the small box before “Negative numbers red”. (A check-mark appears in the box) > Click **OK**. (The number '-9,999.13” appears in cell A1.

Align Left

2. Click **Alighment** tab. In the “Horizoantal” pull-down menu, select **Left**. Click OK. (the numbers move to the left edge of the cell.)

C) IMPRESS

For all your documents IMPRESS is the presentation feature in OpenOffice. A presentation is pages that can be seen on the computer screen as normal and, with the use of a projector, the page can be seen on the wall or on an external screen. Normally, a laptop is used since it has good portability.Impress is like a SLIDE SHOW. One page after the other is projected.A page that is projected is called a SLIDE. The following can be included in a Slide:

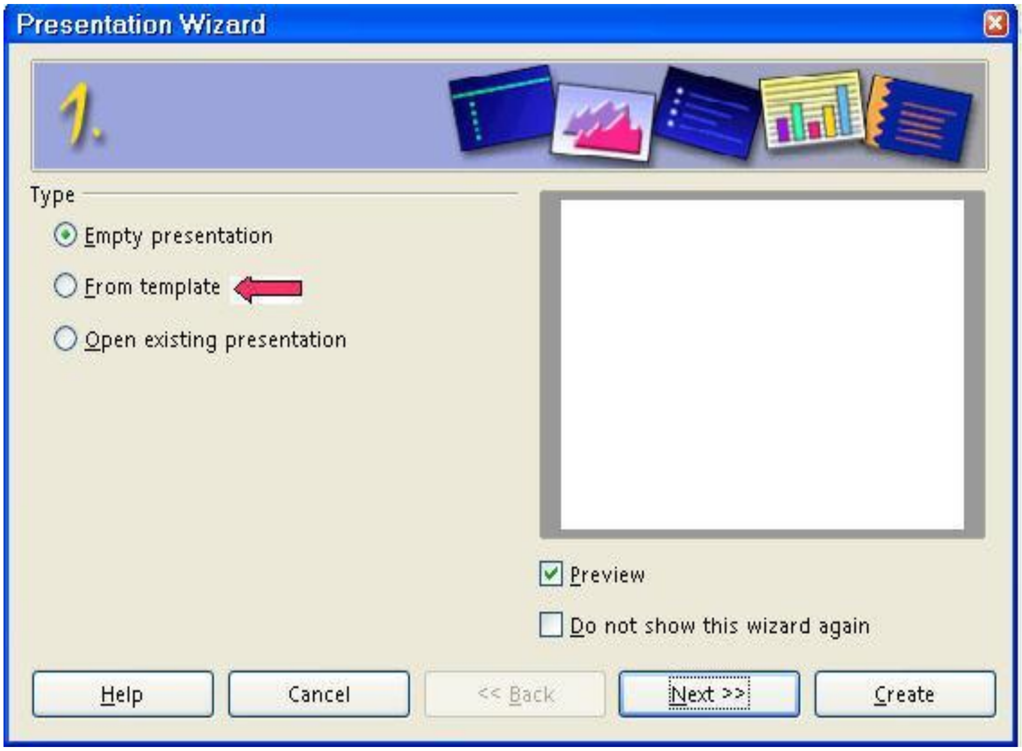
1. Animation

2. 3-D graphics
3. Bullets and Numbered Lists
4. All the normal stuff that is in a word processor (Writer) and in spreadsheets such as bold, colors, text alignment, borders, drawing, templates (master slide) , etc.

Notes can be added to a slide that are not projected but can be seen by the presenter.

Use A Template

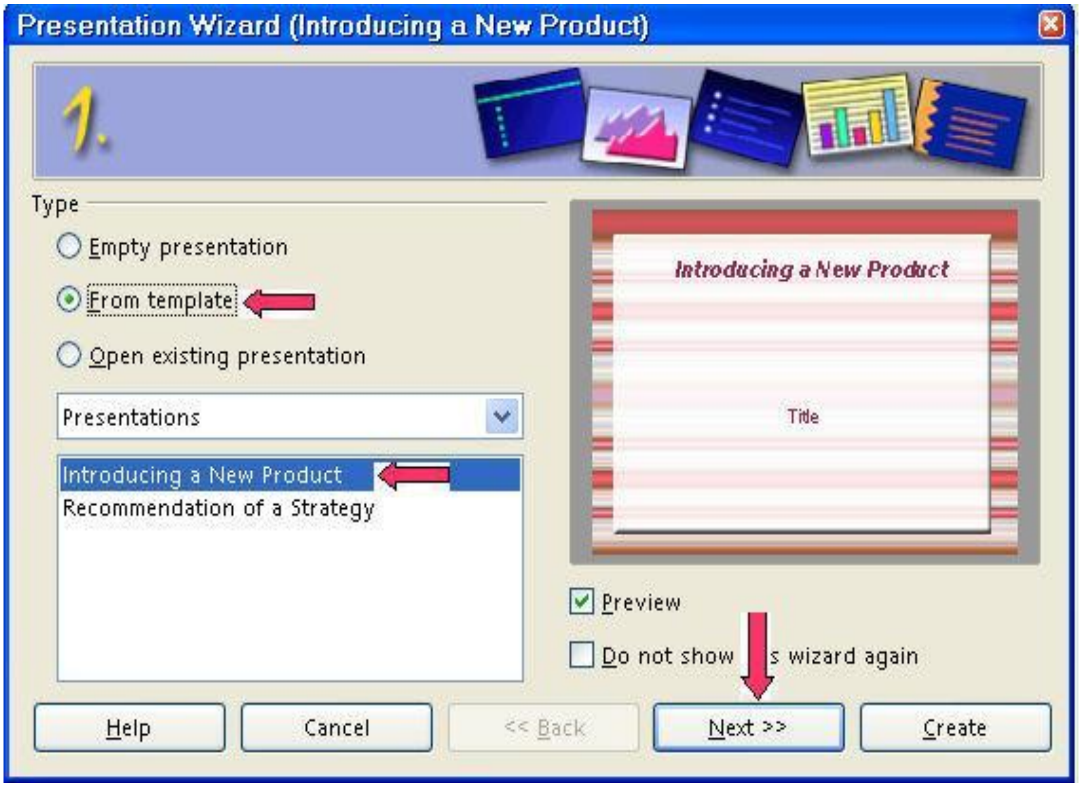
1. OpenOffice has two ready-made templates, **Introducing A New Product** and **Recommendation Of A Strategy.**
2. Click **File > New > Presentation > From Template.** (Part 1 of Window 1 of the “Presentation Wizard” appears.)



Type
● Empty presentation
○ From template
○ Open existing presentation

☑ Preview
□ Do not show this wizard again

3. If not already selected, select **Introducing a New Product**. Click **Next** (Part 2 of Window 1 of the "Presentation Wizard" appears.)



Presentation Wizard (Introducing a New Product)

Type

○ Empty presentation

● From template

○ Open existing presentation

Presentations

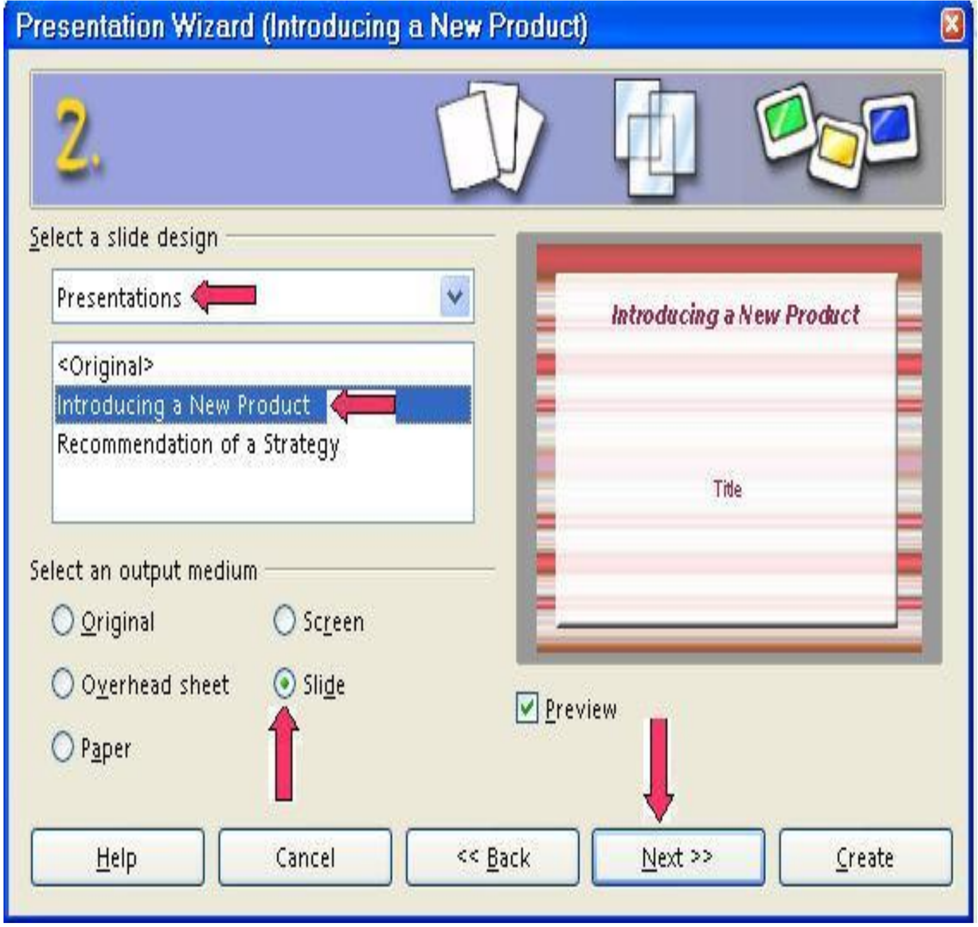
Introducing a New Product

Recommendation of a Strategy

☑ Preview

□ Do not show this wizard again

4. In the dialog box under **<u>S</u>elect a slide design,** click **Presentation** if it is not already selected. (Window 2 of the "Presentation Wizard" appears.



Presentation Wizard (Introducing a New Product)

2.

Select a slide design

Presentations

<Original>
Introducing a New Product
Recommendation of a Strategy

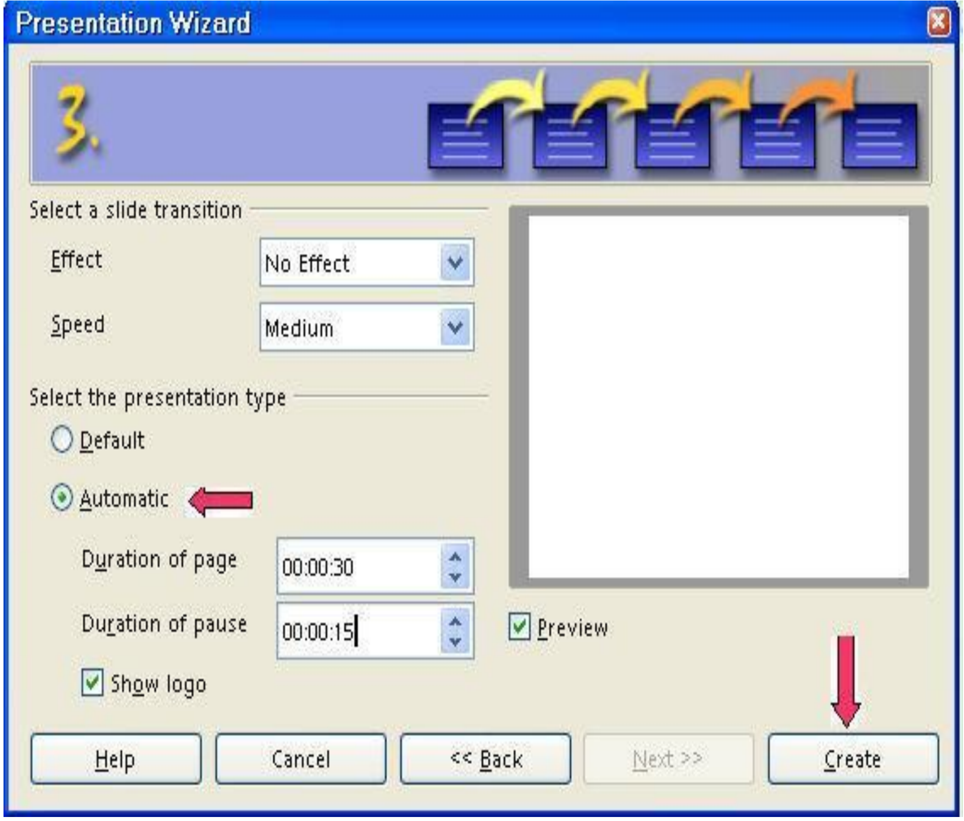
Select an output medium

○ Original ○ Screen
○ Overhead sheet ● Slide
○ Paper

☑ Preview

5. Click **<u>N</u>ext.** (Window 3 of "Presentation Wizard" appears.) Under **Select a slide transition**, use the default for **<u>E</u>ffect** and **<u>S</u>peed**. Under **Select**

Presentation Type**,** click **Automatic.**

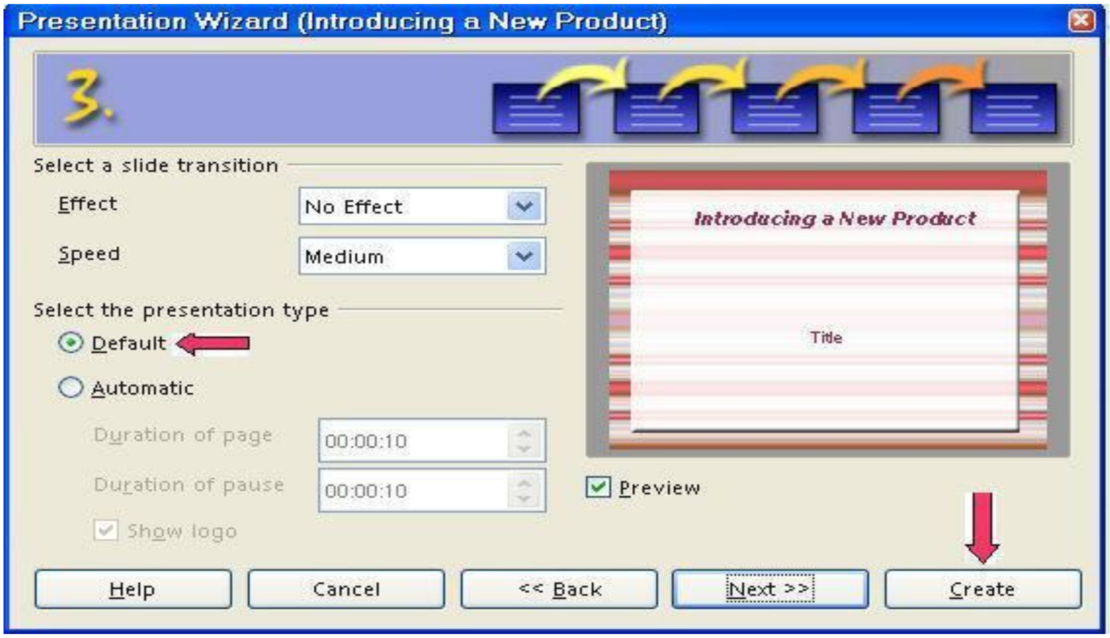


Select a slide transition
Effect: No Effect
Speed: Medium

Select the presentation type
○ Default
● Automatic
Duration of page: 00:00:30
Duration of pause: 00:00:15
☑ Show logo

☑ Preview

6. Note: If you use the default settings, click **Create** and you have your presentation ready for pictures, text. Etc.

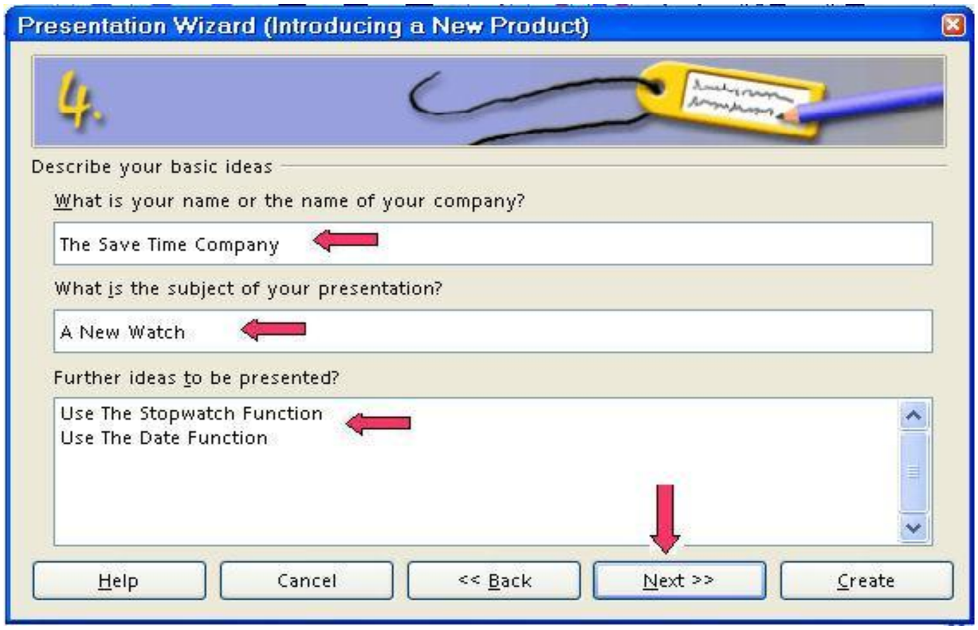

Presentation Wizard (Introducing a New Product)
3.
Introducing a New Product
Title
Help
Cancel
<< Back
Next >>
Create

Select a slide transition
Effect: No Effect
Speed: Medium

Select the presentation type
● Default
○ Automatic
Duration of page: 00:00:10
Duration of pause: 00:00:10
☑ Show logo

☑ Preview

7. Under **Describe your basic ideas**, in the dialog box under **What is your name or the name of your company?**, type **The Save Time Company.** In the dialog box under **What is the subject of your presentation?,** type **A New Watch**. In the dialog box under **Further ideas to be presented**, type **Use the Stopwatch Function** and **Use The Date Function.**
8. Click **Next.** (Window 4 of the 'Presentation Wizard" appears.)



Presentation Wizard (Introducing a New Product)

Describe your basic ideas

What is your name or the name of your company?

The Save Time Company

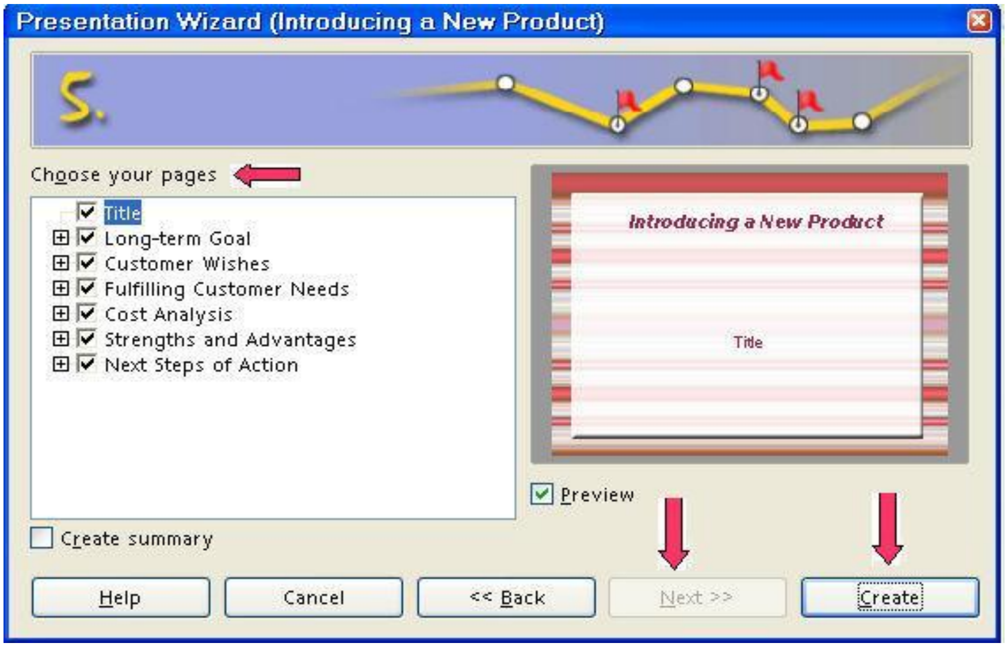
What is the subject of your presentation?

A New Watch

Further ideas to be presented?

Use The Stopwatch Function
Use The Date Function

9. Under **Choose your pages**, select what you want in your presentation or use the default settings. Click **Create.** (Note: "Next" is grayed. "Create" is the next step.**)**


Presentation Wizard (Introducing a New Product)

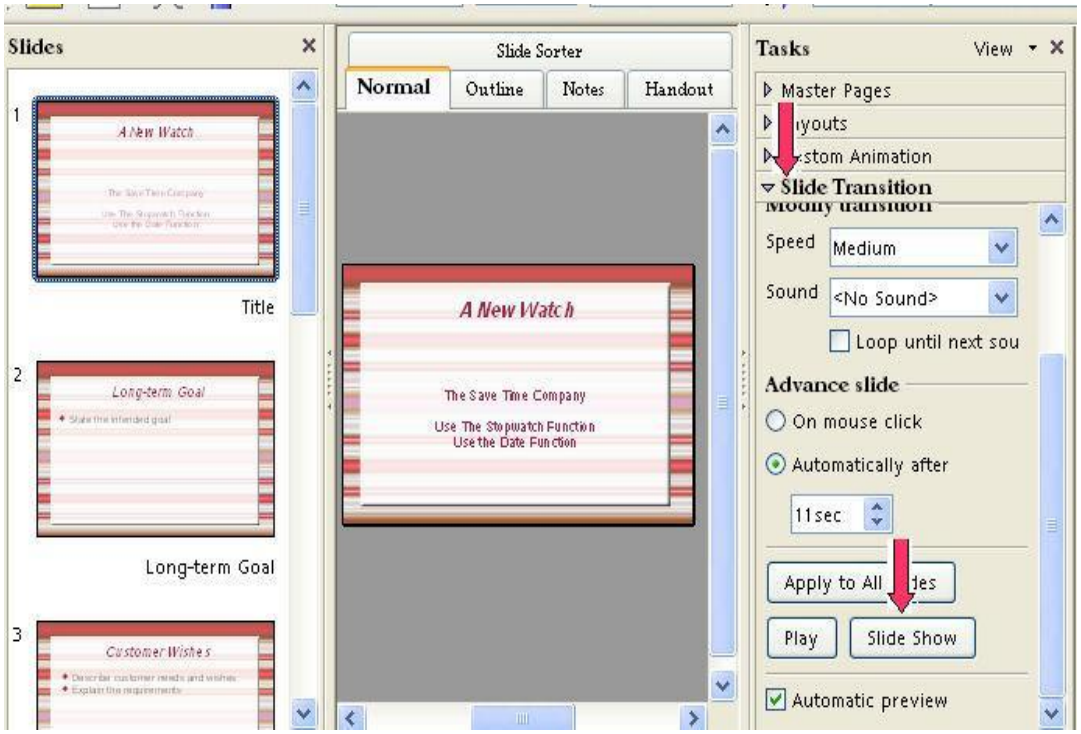
Choose your pages

- ☑ Title
- ⊞ ☑ Long-term Goal
- ⊞ ☑ Customer Wishes
- ⊞ ☑ Fulfilling Customer Needs
- ⊞ ☑ Cost Analysis
- ⊞ ☑ Strengths and Advantages
- ⊞ ☑ Next Steps of Action

☑ Preview

□ Create summary

10. Click **Slide Transition** if it is not already selected. To see your presentation that you just made, click **Slide Show**.


Modify transition

Speed: Medium

Sound: <No Sound>

□ Loop until next sou

Advance slide

○ On mouse click

● Automatically after

11sec

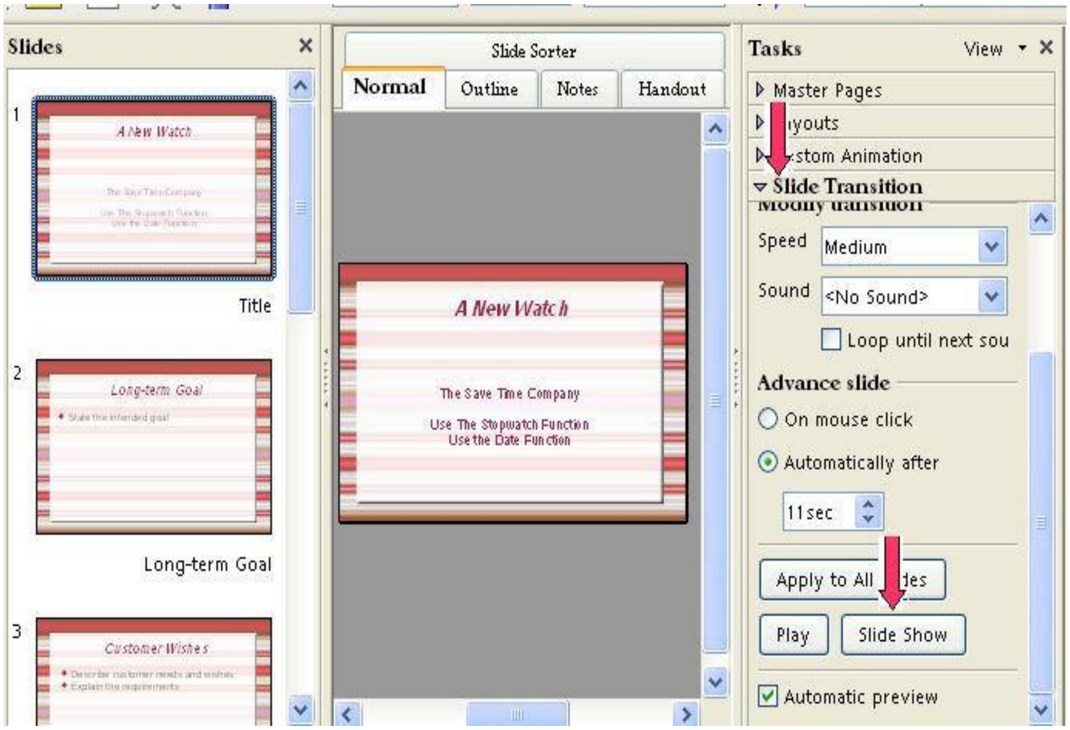
☑ Automatic preview

Save And Close Your Presentation

1. Click **File** > **Save As**. In the **Save in:** pull-down menu box, click **My Documents** (if it is not already selected). In the **Save as type:**" box, click **OpenDocument Presentation (.odp)** (if it is not already selected).
2. In the **File name:** box, type **My Slide Show .**
3. Click **Save.** (" My Slide Show" is saved. The file, "My Slide Show" is still on the screen.)
4. Click **File** > **Close**. (The file, " My Slide Show", is no longer on the screen.)

To Watch "My Slide Show” at a Later Time

1. Click **File** > **Open.** (The "Open" window appears.)
2. In the menu below the "**Look in**:" dialog box, click **My Documents** if it is not already there. In the menu below the **File name:** dialog box, click **My Slide Show**. ("My Slide Show" appears on your screen.)
3. Click **Slide Transition** if it is not already selected. To see your presentation that you just made, click **Slide Show**.



Speed Medium

Sound <No Sound>

☐ Loop until next sou

Advance slide

○ On mouse click

● Automatically after

11sec

☑ Automatic preview

\

PRACTICAL 4

Aim: Write a program to find the nature of the roots as well as value of the roots. However, in case of imaginary roots, find the real part and imaginary part separately

```
#include<conio.h>
#include<stdio.h>
int main()
{
float a, b, c, x1, x2, disc, disc1;
system(“cls”);
printf(“For the quadratic equation of the form: a(x^2) + b(x) + c = 0; \n
Enter the values of the coefficients:\n”);
printf(“Enter the value of a: ”);
scanf( “%f”, &a);
printf(“Enter the value of b: ”);
scanf( “%f”, &b);
printf(“Enter the value of c: ”);
scanf( “%f”, &c);
disc = b * b – 4 * a * c;
if (disc>0)
{
x1 = ( -b + sqrt(disc)) / (2 * a);
x2 = ( -b + sqrt(disc)) / (2 * a);
printf(“The roots are distinct and they are:” );
printf(“%f”, x1);
printf(“%f”, x2);
}

if(disc==0)
{
printf(“The roots are equal and it is:”);
printf(“%f”,(-b/(2 * a)));
}
if(disc<0)
{
disc = - disc;
x1 = -b/(2 * a);
x2 = sqrt(disc1)/(2 * a);
printf(“The roots are complex and they are:”)
printf(“%f”,x1 “ + ” << x2 “i”);
}
getchar();
return 0;
}
```

Output:

```
For the quadratic equation of the form: a(x ^ 2) + b(x) +c = 0;
Enter the values of the coefficients:
Enter the value of a: 1
Enter the value of b: 2
Enter the value of c: 1
The roots are equal and it is: -1
```

PRACTICAL 5

Aim : Write a program, which takes two integer operands and one operator form user, performs the operation and then prints the result. (Consider the operators +,-,*, /% and use switch statement).For example, the input should be in the form: 5 + 3 the output should comes Result = 8

```
#include<conio.h>
#include<stdio.h>
int main()
{
float op1, op2, res;
char ch;
printf(“Enter two numbers:”);
scanf(“%f”, “%f”,&op1,&op2);
printf(“Enter an operator(+,-,*,/,%):”);
scanf(“%c”,&ch);
switch(ch)
{ case '+': res =
op1 +
op2;
break;
case '-': res = op1- op2;
break;
case '*': res = op1 * op2;
break;
case '/': if(op1==0)
printf(“Divide by zero error!!!”);
else
res = op2/op1;
break;
case '%': if(op2==0)
printf(“Divide by zero error!!!”);
else
{ int r,q;
q = op2 / op1;
r = op2 – (q * op1);
res = r;
}
break;
default : printf(“Wrong operator!!”);
}
printf(“The calculated result is :”)
printf("%f",res);
return 0;
}
```

Output:-

```
Enter two numbers : 4 2
Enter an operator(+,-,*,/,%) : /
The calculated result is : 0.5
```

PRACTICAL 6

Aim: Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a program to generate the first n terms of the sequence. For example, for n = 8, the output should be 0 1 1 2 3 5 8 13

```
#include<conio.h>
#include<stdio.h>
int main()
{
int n, c, first = 0, second = 1, next;
printf("Enter the number of terms of Fibonacci series you want");
scanf("%d",&n);
printf("terms of Fibonacci series are :- ");
for ( c = 0 ; c < n ; c++ )
{
if ( c <= 1 )
next = c;
else
{
next = first + second;
first = second;
second = next;
}
printf("%d",next);
}
return 0;
}
```

Output:

```
Enter the number of terms of Fibonacci series you want
8
First 8 terms of Fibonacci series are :-
0112358
13
14
```

PRACTICAL 7

Aim: Write a program to print all the prime numbers between m and n, where the value of m and n is supplied by the user.

```
#include<stdio.h>
#include<conio.h>
void main( )
{ int n, x, flag,ct;
clrscr( );
printf("Enter the n value:”);
scanf(“%d”,&n);
printf("Prime Numbers:");
for( ct=1; ct<=n; ct++)
{
x=2; flag=0;
while(x<=ct/2)
{ if(ct%x==0) {
flag=1; break; }
x++;
} if(flag==0)
printf(“%d”,ct);
}
getch( );
}
```

Output:

```
Enter the n value: 9
Prime number: 2
357
```

PRACTICAL 8

Aim: The number such as 1991, is a palindrome because it is same number when read forward or backward. Write a program to check whether the given number is palindrome or not

```
#include<conio.h>
#include<stdio.h>
int main()
int n, num, digit, rev=0;
printf("\n Input the number (max. 32767):");
scanf("%d",&num);
n=num;
do
{
digit = num %10;
rev = (rev * 10) +digit;
num =num/10;
}
while(num !=0)
printf("The reverse of the number is:");
printf("%d",&rev);
if(n==rev)
printf("The number is palindrome");
else
printf( "\n The number is not a palindrome");
return 0;
}
```

Output:

```
Input Number: 747
The number is palindrome
```

PRACTICAL 9

Aim: A positive integer number IJK is said to be well – ordered if I < J < K. For example number 138 is called well-ordered because the digits in the number (1,3,8) increase from left to right. I.e 1< 3< 8. Number 365 is not well-ordered because 6 digit islarger than 5. Write a program that will find and display the total number of threedigit well-ordered numbers found.

```
#include<stdio.h>
#include<conio.h>
int main()
{ int num, d1, d2, d3, temp, count=0;
printf("Well Ordered Numbers");
for( num = 100; num<200; num++)
{ temp =num;
d3 = temp%10;
temp = temp/10;
d2 = temp%10;
temp = temp/10;
d1 = temp%10;
if(d1<d2 && d2<d3)
{
printf("%d",&num);
count++;
}}
```

Output:

```
Well Ordered Numbers
123 124 125 126 127 128 129 134 135 136
137 138 139 145 146 147 148 149 156 157
158 159 167 168 169 178 179 189
Total number of Well-ordered numbers: 84
```

PRACTICAL 10

Aim : Write a function to compute the highest common factor of integer numbers m and n. Use this function to find the highest common factor of integer numbers a and b.

```
#include<stdio.h>
#include<conio.h>
int calcGCD(int a, int b);
int main()
{ int a, b,gcd;
printf(“\n Enter two numbers, whose GCD/HCF is to be found:”);
scanf(“%d”,”%d”,&a,&b);
gcd=calcGCD(a,b);
printf(”\n GCD or HCF of given numbers”);
printf(“%d”,gcd);
getchar();
return 0;
}
int calcGCD()int a, int b)
{ int num, den, GCD, r;
if(a > b)
{
num = a;
den = b;
} else
{
num = b;
den =a;
}
while(den > 1)
{
r = num % den;
if(r == 0)
{
GCD = den;
break;
} else
{
num = den;
den =r;
}} if(den == 1)
GCD = 1;
```

```
return getch();
}
```

Output:

```
Enter two numbers, whose GCD/HCF is to be found: 325 45
GCD or HCF of given numbers: 5
```

PRACTICAL 11

Aim: **Given the marks (out of 100) obtained by each student in a test of a class with n students. Write a program to obtain the following information:**
(a) minimum and maximum marks score
(b) average score of the class, and (c) number of students whose score is greater than class's average score

```
#include<stdio.h>
#include<conio.h>
const int size = 30;
int main()
{
double marks[size], sum=0, avgMarks=0,max=0,min=0, count=0;
for (int =0; i<size; i++)
{
 printf("Enter marks of student");
scanf("%f",marks[i]);
sum+=marks[i];
if(marks[i] > max)
max= marks[i];
if(i==0) min = marks[0];
else if (marks[i] < min)
min = marks[i];
} avgMarks =
sum/size;
for (int i=0; i<size; i++)
{
if(marks[i]; i<size; i++)
count++;
}
printf( "Maximum marks obtained:" );
printf("%f",max);
printf( "Minimum marks obtained:" );
printf("%f",min);
printf( "average marks obtained:" );
printf("%f",avgMarks);
printf("Number of students having score more than average score of the class:");
printf("%d",count);
return 0;
}
```

Output:

Marks array contains following data

```
86 58 45 78 45 89 35 67 23 56 89 99
100 67 36 75 46 23 76 47 69 78 36
57 68 67 97 42 95 86
```

And the program produces following output

```
Maximum marks obtained: 100
Minimum marks obtained: 23
Average marks obtained: 64.433
Number of students having score more than average score of the class: 17
```

PRACTICAL 12

<u>Aim</u>: Write a program to multiply matrix Am×n by B p×q, given that n = p.

```
#include<stdio.h>
#include<conio.h>
void main()
{ clrscr();
int a[10][10],b[10][10],c[10][10],m,n,o,p,i,j;
printf("Enter number of rows of A: ");
scanf(“%d”,&m);
printf("Enter number of coloumns of A: ");
scanf(“%d”,&n);
printf("Enter elements of matrix A: ");

for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{ printf("Enter element “);
scanf(“%d”,&a[i][j]);
}}
printf("Enter number of rows of B: ");
scanf(“%d”,&o);
printf("Enter number of columns of B: ");
scanf(“%d”,&p);
printf("Enter elements of matrix B: “);
for(i=0;i<o;i++)
{
for(j=0;j<p;j++)
{ printf("Enter element b");
scanf(“%d”,&b[i][j]);
}}
printf("Displaying Matrix A: ");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
printf(“%d”,a[i][j]);
}
} printf("Displaying Matrix B: ");
for(i=0;i<o;i++)
{
```

```
for(j=0;j<p;j++)
{ printf(“%d”,b[i][j]);
}
} if(n==o)
{
for(i=0;i<m;i++)
{
for(j=0;j<p;j++)
{ c[i][j]=0;
for(int k=0;k<n;k++)
{ c[i][j]=c[i][j]+a[i][k]*b[k][j];
}}}
printf("Matrix A*Matrix B=Matrix C: “);
for(i=0;i<m;i++)
{
for(j=0;j<p;j++)
{ printf(“%d”,c[i][j]);
}
}} else
printf("Multiplication not possible");
getch();
}
```

PRACTICAL 13

Aim: Write a program to sort a list of n integer numbers in descending order using bubble sort method

```
#include<conio.h>
#include<stdio.h>
int main()
{
system("cls");
int A[25],i,j,tmp,size;
printf("\n Enter number of elements in the array:");
scanf("%d",&size);
printf("\n Enter the elements of array:");
for(i=0; i<size;i++)
scanf("%d",&A[i]);
for(i=0; i<size; i++)
{
int last = (size-1) -i;
for(j=0; j<last; j++)
if(A[j] < A[j+1])
{ tmp=A[j];
A[j]=A[j+1];
A[j+1]=tmp;
}
}
printf( "\n Array in descending order is: \n");
for(i=0; i<size;i++)
printf("%d",A[i]);
return 0;
}
```

Output:

```
Enter the number of elements in the array: 5
Enter the elements of array:
3 2 1 4 5 2
Array in descending order is:
12345
```

PRACTICAL 14

Aim: Write a program to perform the addition and multiplication of two complex numbers using structures.

```
#include <stdio.h>
#include <stdlib.h>
#include<conio.h>

struct complex
{
  int real, img;
};

int main()
{
  int choice, temp1, temp2, temp3;
  struct complex a, b, c;

  while(1)
  {
    printf("Press 1 to add two complex numbers.\n");
    printf("Press 2 to multiply two complex numbers.\n");
    printf("Press 3 to exit\n");
    printf("Enter your choice\n");
    scanf("%d",&choice);

    if( choice == 5)
      exit(0);

    if(choice ==1 || choice == 2)
    {
      printf("Enter a and b where a + ib is the first complex number.");
      printf("\na = ");
      scanf("%d", &a.real);
      printf("b = ");
      scanf("%d", &a.img);
      printf("Enter c and d where c + id is the second complex number.");
      printf("\nc = ");
      scanf("%d", &b.real);
      printf("d = ");
      scanf("%d", &b.img);
    }
    if ( choice == 1 )
    {
      c.real = a.real + b.real;
      c.img = a.img + b.img;
    if ( c.img >= 0 )
      printf("Sum of two complex numbers = %d + %di",c.real,c.img);
      else
      printf("Sum of two complex numbers = %d %di",c.real,c.img);
    }

    else if ( choice == 2 )
    {
      c.real = a.real*b.real - a.img*b.img;
      c.img = a.img*b.real + a.real*b.img;
  if ( c.img >= 0 )
      printf("Multiplication of two complex numbers = %d + %di",c.real,c.img);
      else
      printf("Multiplication of two complex numbers = %d %di",c.real,c.img);
    }
    else
      printf("Invalid choice.");

    printf("\nPress any key to enter choice again...\n"); }  }
```

PRACTICAL 15

Aim: Write a program using pointers to compute the sum of all elements stored in an array

```
#include<stdio.h>
#include<conio.h>
void main() {
int numArray[10];
int i, sum = 0;
int *ptr;

printf("\nEnter 10 elements : ");
for (i = 0; i < 10; i++)
scanf("%d", &numArray[i]);
ptr = numArray; /* a=&a[0] */

for (i = 0; i < 10; i++) {
sum = sum + *ptr;
ptr++; }
printf("The sum of array elements : %d", sum); }
```

Output

```
Enter 10 elements: 11 12 13 14 15 16 17 18 19 20
The sum of array elements is 155
```