

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO1</b>	Use state of art tools to develop professional documents, spread sheets, web layouts and presentations in various assignments					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
						<b>Total</b>
						<b>/10</b>

### **Lab No. 1**

### **Learning Objectives:**

By reading this manual, students will be able to:

- Identifying the names and purposes of different computer hardware components
- Become familiar with the different commands in command prompt

### **Computer Systems:**

A computer system consists of all the components (hardware and software) used to take information from the computer users, process it and generate output. It is generally broken down into five basic components:

**Arithmetic Logic Unit (ALU):** The arithmetic and logic unit (ALU) of a computer system is the place where the actual calculations are performed and all comparisons (decisions) are made.

**Central Processing Unit (CPU):** The control Unit and ALU are jointly known as the Central Processing Unit.

The CPU is the brain of any computer system. In a human body, all major decisions are taken by the brain and the other parts of the body act as directed by the brain. Similarly, in a computer system, all major calculations and comparisons are made inside the CPU and the CPU is also responsible for activating and controlling the operations of other units of a computer system.

**Main Memory:**

The main memory of the computer is also known as RAM, standing for Random Access Memory. It provides temporary application data storage. Instead of having to go back and find information on the hard drive every time a computer needs a piece of data or instructions, the computer temporarily stores frequently used files on RAM, making them easier to find. When the computer shuts down, all the data in RAM is erased or transferred to the hard drive, making room for new data when the computer begins operating again.

**Secondary Storage:**

The most common form of external memory is a hard disc which is permanently installed in the computer and typically has a capacity of hundreds of megabytes. It is sometimes called backing store or secondary memory. It stores programs or data not actively being used by the other units. It retains the information even when the computer's power is turned off.

**Input Unit:**

This “receiving” section obtains information (data and computer programs) from input devices such as keyboards, touch screens and mouse devices.

**Output Unit:**

This “shipping” section takes information that the computer has processed and places it on various output devices to make it available for use outside the computer. Most information that’s output from computers today is displayed on screens, printed on paper, played as audio or video on PCs. “receiving” section obtains information (data and computer programs) from input devices such as keyboards, touch screens and mouse devices.

**Command Prompt Commands:**

The command prompt, a purely text-driven interface, is still present at the core and it contains many options that are not otherwise accessible without third-party software.

A command prompt is a non-graphical interface that allows you to interact with your operating system. At the command prompt, you enter commands by typing their names followed by options and arguments. Most modern computers use a graphical user interface (GUI), which allows users to more intuitively access programs and documents.

**Lab Activity:**

To access the command prompt in Windows, In the “search” field, enter cmd or command prompt.

**Note:** You have to verify each command given below using command prompt and make lab report of your lab activity. Some Commands are:

<b>Command</b>	<b>Description</b>
<b>dir</b>	To view the contents of a directory, type “dir”. This command will list all the files and directories within the current directory. It is analogous to clicking on a Windows folder to see what's inside.
<b>cd</b>	<p>It is frequently useful to know in which directory you are currently working. To find out, type “cd” at the command prompt.</p> <p>To change directories, use the “cd” command with the name of a directory e.g. “c: \ &gt;cd name of directory”.</p> <p>To return to the previous directory, use the cd command, but this time followed by a space and two periods e.g. “c: \ &gt;cd ..”.</p>
<b>mkdir</b>	To create a new directory, use the command “mkdir”. The following command creates a directory named “hello”, which you can use to store all your files associated with the Hello World assignment.
<b>move</b>	<p>The move command will move a file to a different location. Now, create a new directory using mkdir and move that file into the hello directory using the “move” command. You can also use move to rename a file. Simply specify a new filename instead of a directory name. Suppose you accidentally messed up the upper and lower case and had saved Hello as hello. Use two move commands to fix it e.g.</p> <pre>C:\ &gt; move hello temp C:\ &gt; move temp Hello C:\ &gt; dir</pre>
<b>copy</b>	To make a copy of a file, use the copy command.
<b>del</b>	The del command deletes a file.
<b>cls</b>	If you have lots of text in your Command Prompt window, this command will clear it. The history of previous commands typed in that session will be maintained.
<b>help</b>	If you want a full list of the commands available to you, just type "help" and they will be listed.
<b>ipconfig</b>	This command is a fast way of determining your computer's IP address and other information, such as the address of its default gateway.
<b>ping</b>	Type “ping google.com” and Windows will send packets to Google.com. Google will respond and let you know it's received them. You'll be able to see if any packets didn't make it to Google.com, perhaps you're experiencing packet loss and how long it took you to hear back, perhaps the network is saturated and packets are taking a while to reach their destinations.



<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO1</b>	Use state of art tools to develop professional documents, spread sheets, web layouts and presentations in various assignments					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
						<b>Total</b>
						<b>/10</b>

## **1 LAB 2(INTRODUCTION TO DOCUMENTATION TOOLS)**

---

### **1.1 GOALS OF THE DAY:**

- Introduction to Microsoft Word

## **2 SECTION A**

---

### **2.1 INTRODUCING THE WORD USER INTERFACE:**

The Word user interface provides intuitive access to all the tools you need to develop a sophisticated document tailored to the needs of your audience. When you're working with a document, it is displayed in an app window that contains all the tools you need to add and format content.

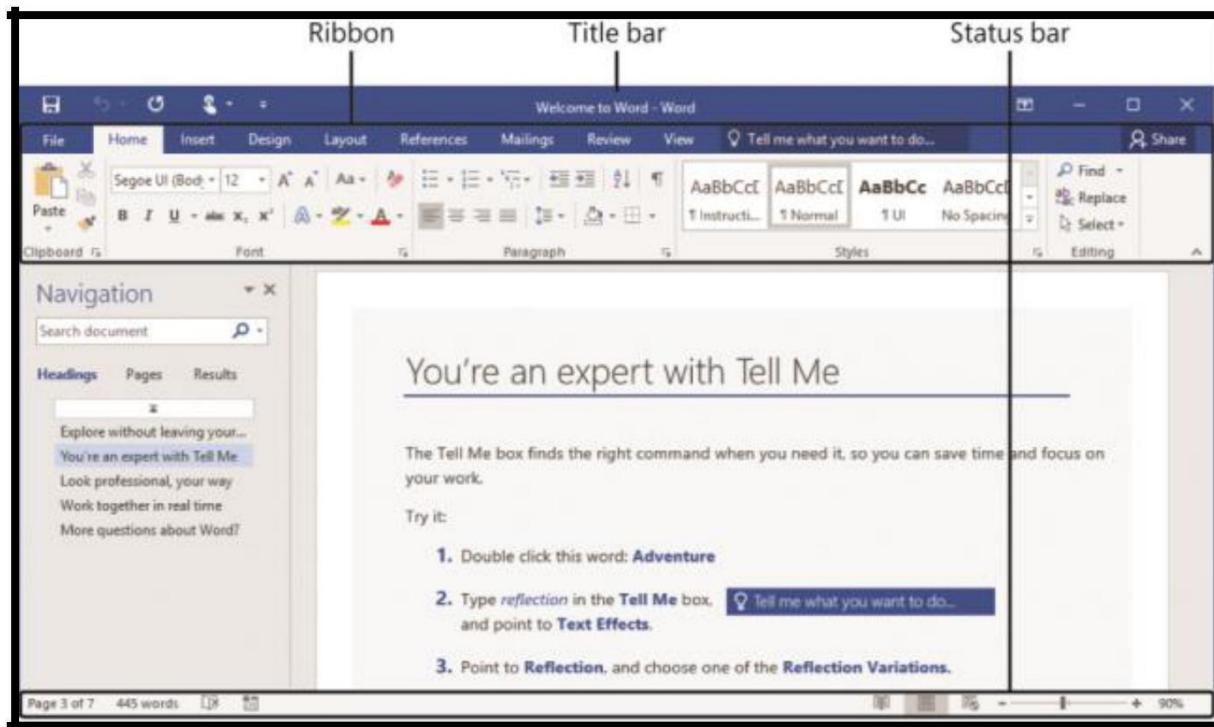


Figure 1: A document displayed in the app window

## Identify app window elements

The Word app window contains the elements described in this section. Commands for tasks you perform often are readily available, and even those you might use infrequently are easy to find.

- Ribbon
- Title bar
- Status bar

1. Using the technique that is appropriate for your operating system, start Word.
2. When the **Start** screen appears, press the **Esc** key to create a new blank document.

The **ribbon** is located below the title bar. The commands you'll use when working with a document are gathered together in this central location for efficiency. (Figure 2)

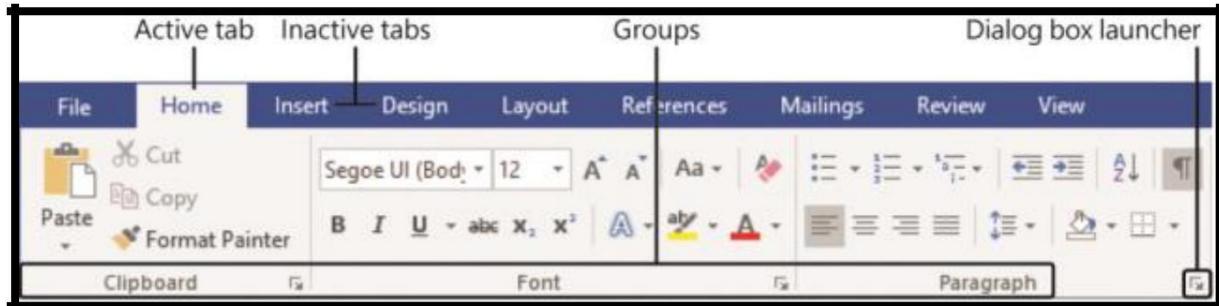


Figure 2: Your ribbon might display additional tabs

Across the top of the ribbon is a set of tabs. Clicking a tab displays an associated set of commands arranged in groups. Commands related to managing Word and documents (rather than document content) are gathered together in the **Backstage view**, which you display by clicking the File tab located at the left end of the ribbon. Commands available in the Backstage view are organized on named pages, which you display by clicking the page tabs in the coloured left pane. You redisplay the document and the ribbon by clicking the Back arrow located above the page tabs. (Figure 3)

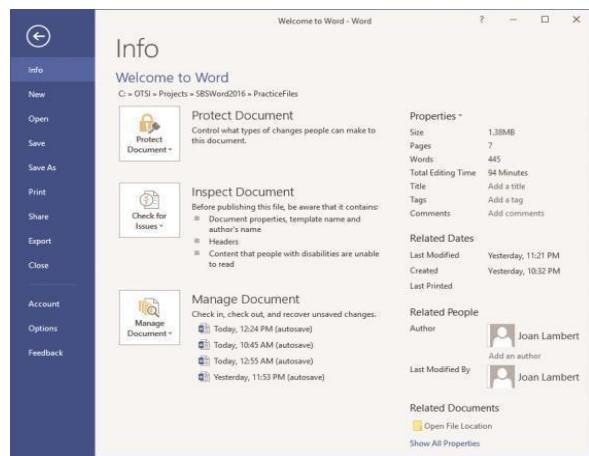


Figure 3: Backstage view

On each tab, buttons representing commands are organized into named groups. You can point to any button to display a **ScreenTip** that contains the command name, a description of its function, and its keyboard shortcut (if it has one) (Figure 4).

Some **buttons include an arrow**, which might be integrated with or separate from the button. To determine whether a button and its arrow are integrated, point to the button to active it. (Figure 5)

**Maximizing and minimizing the word** interface impacts on the ribbon and title page. You can see Figure 6 for 1024 pixels wide screen.

To auto-hide ribbon and other customization options, click on fourth-left icon on the extreme top-left side of word application. See Figure 8 for details. It is also called **Ribbon Display Options** menu.

# University Of Engineering and Technology, Lahore

## Computer Engineering Department



Figure 4: ScreenTips can include the command name, keyboard shortcut, and description



Figure 6: At 1024 pixels wide, most button labels are visible

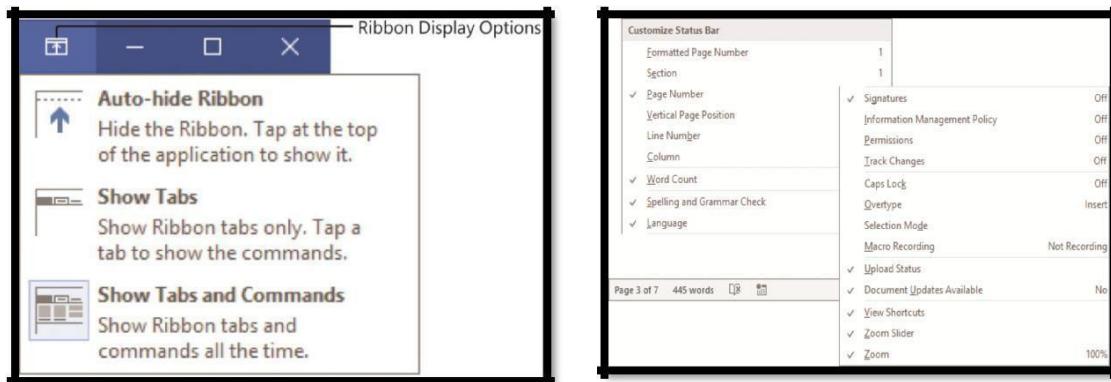


Figure 7 You can specify which items to display in status bar

Figure 8: Ribbon display options menu

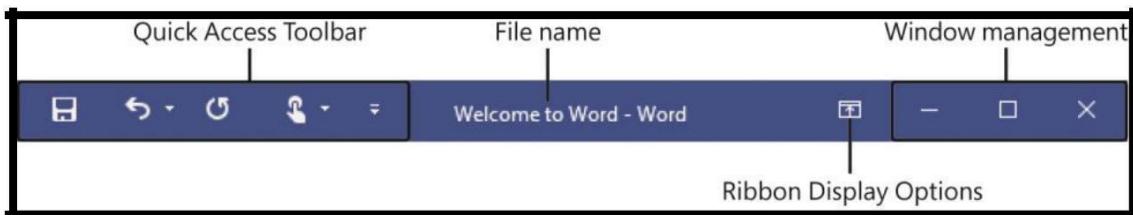


Figure 9: The title bar elements are always on the left end, in the centre, and on the right end of the title bar

At the top of the app window, **title bar** displays the name of the active file, identifies the app, and provides tools for managing the app window, ribbon, and content. (Figure 9)

The **Quick Access Toolbar** at the left end of the title bar can be customized to include any commands that you want to have easily available. The default Quick Access Toolbar in the Word app window displays the Save, Undo, and Redo/Repeat buttons. On a touchscreen device, the default Quick Access Toolbar also includes the Touch/Mouse Mode button. (Figure 9).

Across the bottom of the app window, the **/status bar** displays information about the current document and provides access to certain Word functions. You can choose which statistics and tools appear on the /status bar. Some items, such as Document Updates Available, appear on the /status bar only when that condition is true. (Figure 7)

The **View Shortcuts toolbar**, Zoom Slider tool, and Zoom button are at the right end of the /status bar. These tools provide you with convenient methods for changing the display of document content. (Figure 10).

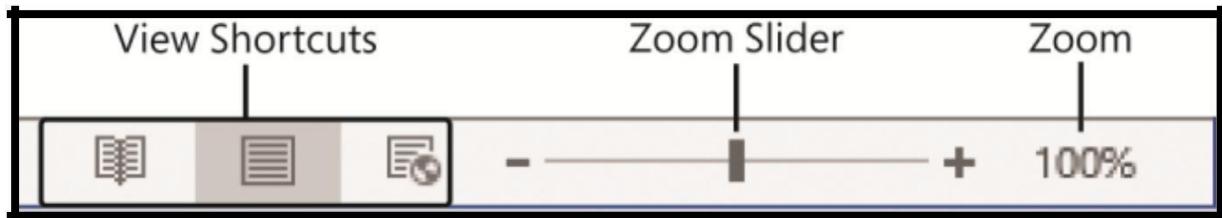


Figure 10: You can display different content views and change the magnification from the status bar

1. Right-click the status bar to display the Customize status Bar menu. A check mark indicates each item that is currently enabled.
2. Click to enable or disable a status bar indicator or tool. The change is affected immediately. The menu remains open to permit multiple selections.
3. When you finish, click away from the menu to close it.

## 2.2 CREATE AND MANAGE DOCUMENTS:

You can use Word 2016 to create many different types of documents, for many different purposes. Word is widely used in schools, businesses, and organizations of many kinds to create letters, newsletters, reports, resumes, and other documents that contain text. Word provides a lot of flexibility in document design, so you can also create documents that contain images and content that doesn't fit a standard sheet of paper, such as trifold brochures, greeting cards, business cards, certificates, and signs.

### 2.2.1 Create documents:

When creating a document in Word, you can create a blank document of the default file type or create a document based on one of the templates provided with Word. Each template incorporates specific design elements such as fonts and colours. Many templates also include typical information that you can modify or build on to create a useful document. (Figure 11)

When you base a new document on a template, that template is said to be attached to the document. The styles defined in the attached template appear in the Styles pane so that you can easily apply them to any content you add to the document. You can change the document template by attaching a different one.

**Formatting:** Most templates contain formatting information, which in addition to styles can include page layout settings, backgrounds, themes, and other types of formatting. A template that contains only formatting defines the look of the document; you add your own content.

**Building blocks:** Some templates make custom building blocks, such as headers and footers or a cover page, available for use with a document. They might also include AutoText, such as contact information or standard copyright or privacy paragraphs.

**Custom tabs, commands, and macros:** Sophisticated templates might include custom ribbon tabs or toolbars with commands and macros that are specific to the purposes of the template. A **macro** is a recorded series of commands that permits a user to perform a process with the click of a button. The topic of macros is beyond the scope of this book; for information, refer to Word Help.

## University Of Engineering and Technology, Lahore Computer Engineering Department

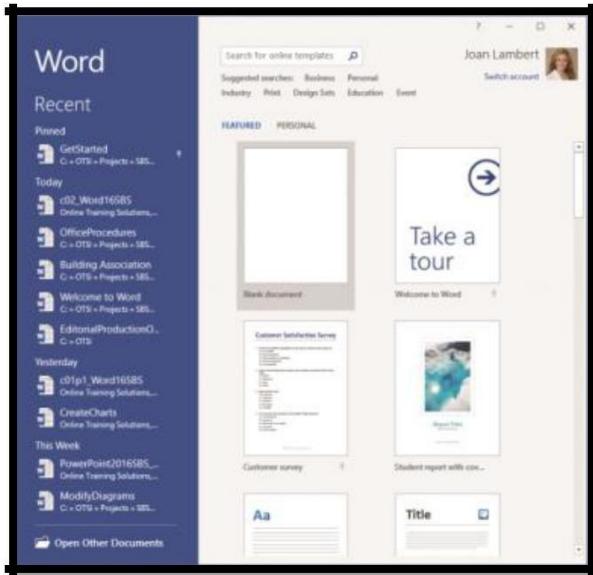


Figure 11: The Start screen appears by default

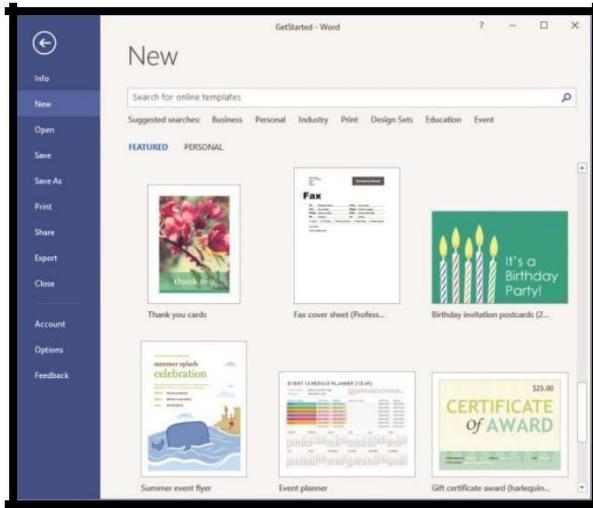


Figure 12: Templates

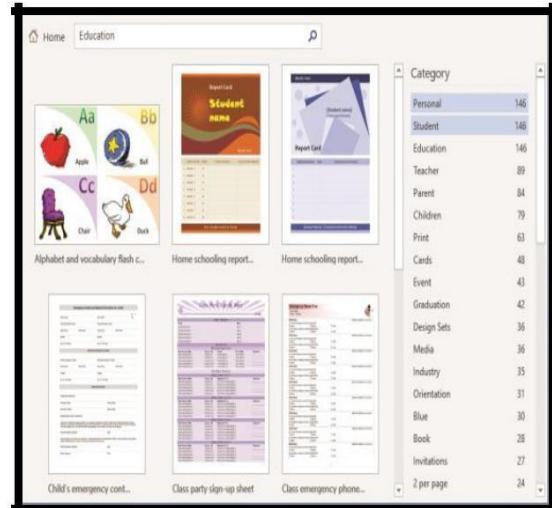


Figure 13: Templates (Category search)

### 2.2.2 Writing text in Word using Tabs:

**File tab:** File menu has many important features which includes save which is used to save documents, open used to open saved documents, new used to open new blank document and print used to print documents.

**Insert tab:** It has many options like inserting table, pictures, smart art, charts, header and footer.

**Home tab:** It is used to change the appearance of text. It can make text bold, italics and underlined. It is also used to change the colour spacing and alignment of text.

**Design tab:** It contains command related to changing the overall appearance of a document such as page borders, page colours and watermark etc.

Exiting Ms Word:

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

You can exit word by clicking the close button in the upper right corner of title bar. Or by selecting file menu and click on close option.

You can use options explained in Figure 14 and Figure 15 for applying fonts, colours and text effects on your writing. You can also use shortcut keys for quick tasking. For example, for saving a document press **Ctrl** key with **S** key. (Figure 16)

Name	Description
<b>B</b> Bold	Makes the selected text bold ( <i>example</i> ).
<b>I</b> Italic	Italicizes the selected text ( <i>example</i> ).
<b>U</b> Underline	Draws a line under the selected text ( <i>example</i> ). Click the arrow on the button to select the type of underline.
<b>abc</b> Strikethrough	Draws a line through the middle of the selected text ( <i>example</i> ).
<b>X<sub>2</sub></b> Subscript	Creates small letters below the text baseline ( <i>example</i> ).
<b>X<sup>2</sup></b> Superscript	Creates small letters above the line of text ( <i>example</i> ).
<b>A</b> Text Effects and Typography	Applies a visual effect (such as a shadow, glow, or reflection) to the selected text.
<b>Aa</b> Change Case	Changes the selected text to uppercase, lowercase, or other common capitalizations.

Figure 14: Font-group and text effects

Name	Description
<b>Align Left</b>	Aligns the text at the left margin, producing a ragged right edge. This is the default alignment.
<b>Center</b>	Centers the text between the left and right margins, producing ragged left and right edges.
<b>Align Right</b>	Aligns the text at the right margin, producing a ragged left edge.
<b>Justify</b>	Aligns the text at both the left and right margins, producing even left and right edges. To accomplish this, Word adjusts the spacing between words, expanding or shrinking the spaces as needed.

Figure 15: Alignment options

Ctrl+P	Printing a document
Ctrl+V	Paste
Ctrl+X	CUT
Ctrl+C	Copy
Ctrl+Z	Undo an action
Ctrl+Y	Redo an action
Ctrl+S	Saving a document

Figure 16: Widely used short-cut keys

### **3 LAB TASKS (TO BE PERFORMED IN LAB)**

#### **3.1 WRITE AN APPLICATION TO THE CHAIRMAN OF DEPARTMENT FOR THE CHANGE IN DISCIPLINE.**

##### **Steps:**

1. Start Ms Word from the start button as mentioned above. And open new document by clicking on new option and selecting blank document. (Figure 18)
2. Start typing in the blank working area according to the given format.
3. Appearance of document can be changed. You can make the heading bold, italics and underlined. Select the text you want to format. On the Home tab from the font group, font size and style can be changed. (Figure 19)
4. Font size can be changed by clicking the font size arrow and selecting the desired font. (Fig 20)
5. Font style can be changed by clicking the font arrow and selecting the desired one. (Figure 17)
6. Change Indentation and alignment of the text if needed. The text in the application can be left, right, centre and justify. Indention of the paragraph can be increased or decreased. All this can be done from Home tab in the paragraph group. (Figure 21)
7. After all the formatting, document can be saved. To save a document click on File tab and then select Save As. Click on the browse button. (Figure 22 and Figure 23)
8. The save as dialog appears. Enter the name of the file in the bar next to File Name. And in the Save As Type box enter the format in which you want to save the file. Then press OK, file will be saved.

**Sample**

*University of Engineering  
and Technology*

**My First Day at University**

I was very excited at first day of my university. Because I have chosen best engineering university of Pakistan. A night before the first day, I prepared my clothes and other accessories. Next day, when I entered UET, I came across a new world. It was the orientation day. I went to my Computer Engineering department but we were asked to visit the university till 10 am then to reach the auditorium. I visited the university.

Senior students were ragging the new ones. Some seniors also tried to rag me but I was careful so I did not allow them to tease me. After visiting the UET, I went to auditorium. After some wait, orientation ceremony began. We were oriented about university, its curriculum, its rules and regulations. Ceremony continued for three hours. After that, we were visited university by bus. Then, I returned home. It was a good day.

# University Of Engineering and Technology, Lahore

## Computer Engineering Department



Figure 17: Changing font

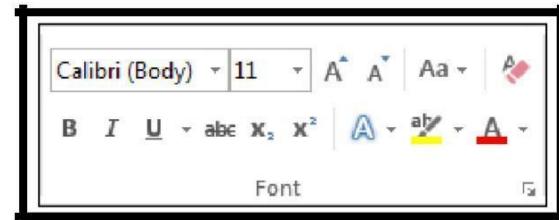


Figure 19: Font group

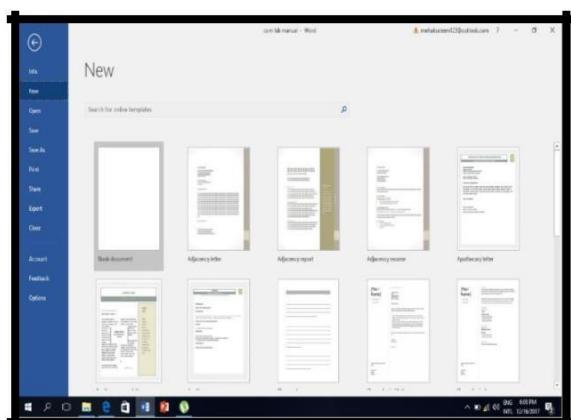


Figure 18: New document

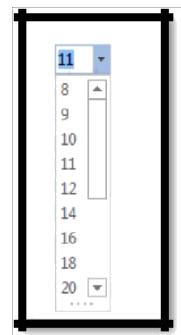


Figure 20: Changing font

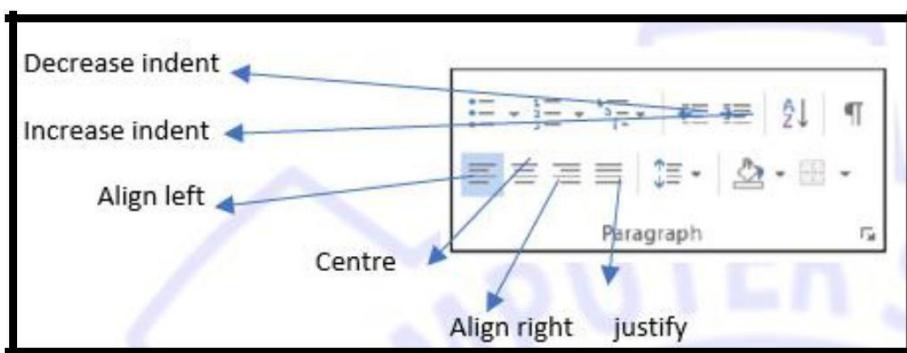


Figure 21: Paragraph group

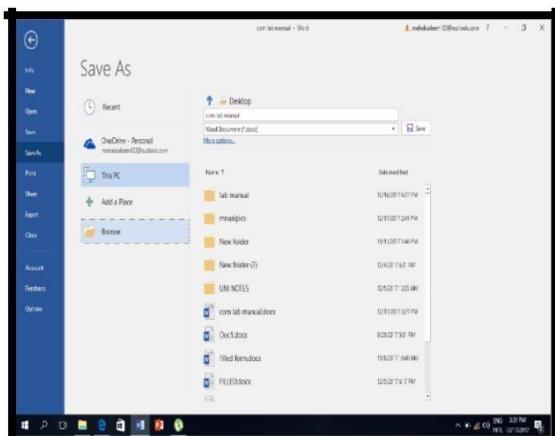


Figure 22: Saving a document

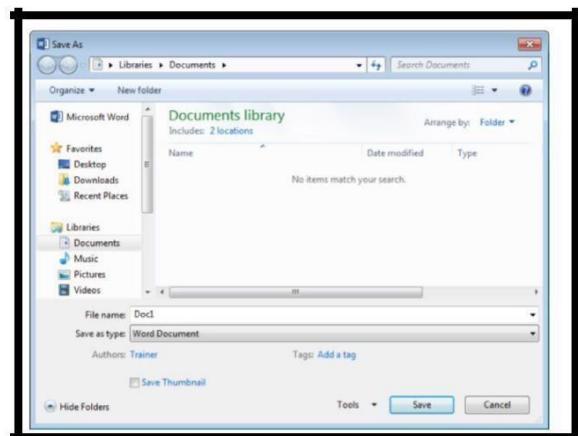
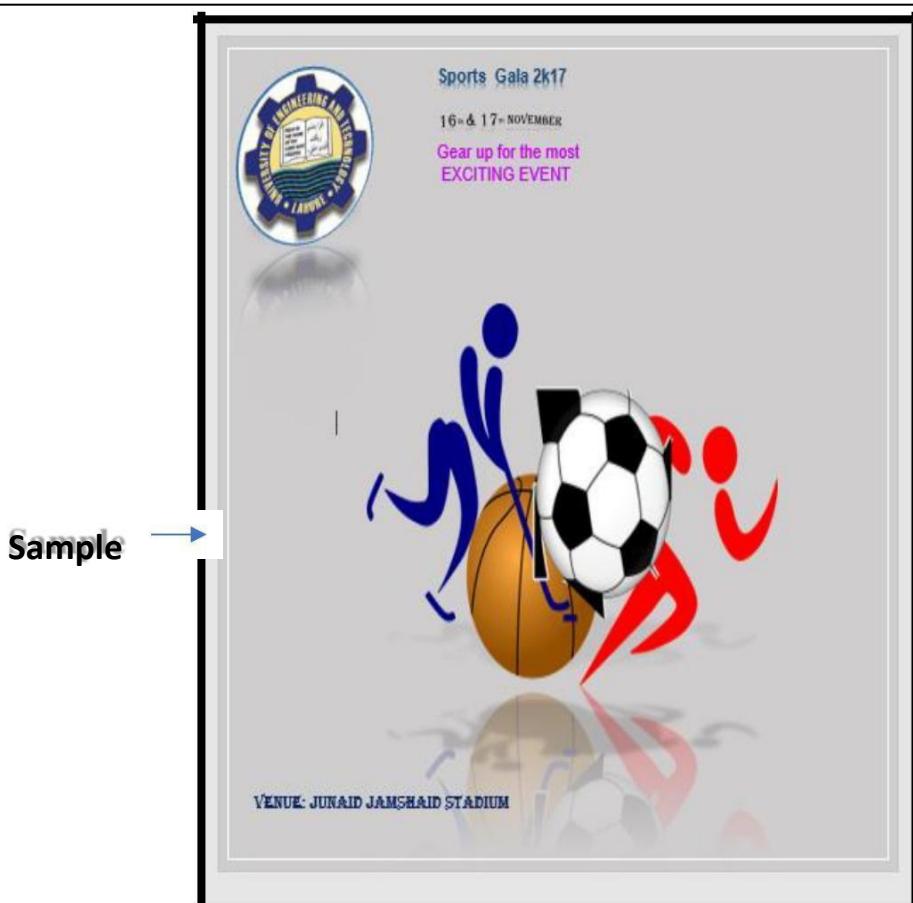


Figure 23: Saving your file

## CREATE A POSTER FOR SPORTS GALA FOR CE DEPARTMENT

1. Open a new blank Word document.
2. Set the paper width and height to 22" by 22". For this click on the layout tab and select size. From drop-down list select **more paper sizes**. (Figure 24)
3. Page Setup dialog box appears. Change the width and height to 22" and click OK. (Figure 25)
4. First, insert the picture related to the given topic. Click Insert tab and select pictures options. (Figure 26)
5. Insert pictures dialog box appears. Select the desired picture you want to insert and then click insert. (Figure 28)
6. To format the picture, select the picture and then select Format Picture from drop-down list. (Figure 27)
7. Format picture pane appears, which helps to crop the picture, changing picture colours, applying 3-D rotation and many more. (Figure 30)
8. Text can be inserted by selecting the picture and then select wrap text option from the drop-down list or from the layout options. (Figure 29)
9. Text colours can be changed from the theme colours available in Font group from Home tab. (Figure 31)
10. Text highlight colour and text effects can also be applied from the font group. (Fig 32)
11. On Insert tab, click on arrow and select desired Word Art. (Figure 34)
12. Colour of background can also be changed. Go to Design tab and select Page Colour and change the colour to the desired one.
13. Check the print preview of the poster. To do this, select the File tab and then select Print option (Figure 35)
14. After checking the print preview, save the poster with desired name. (Figure 36)



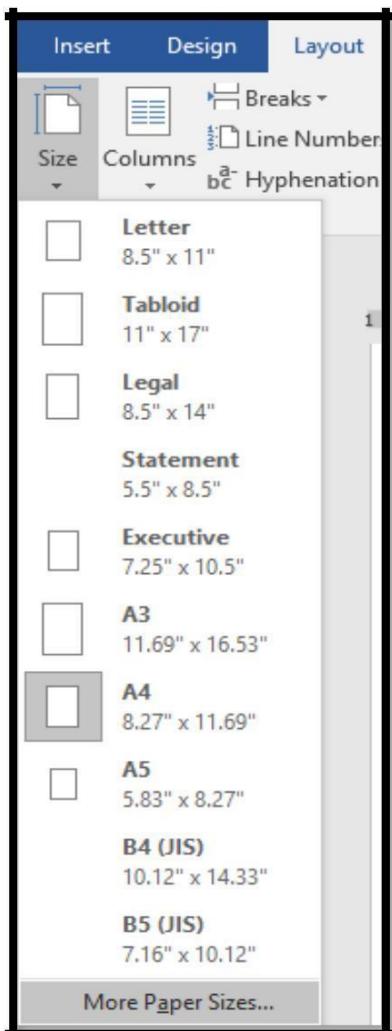


Figure 24: Paper types in Word

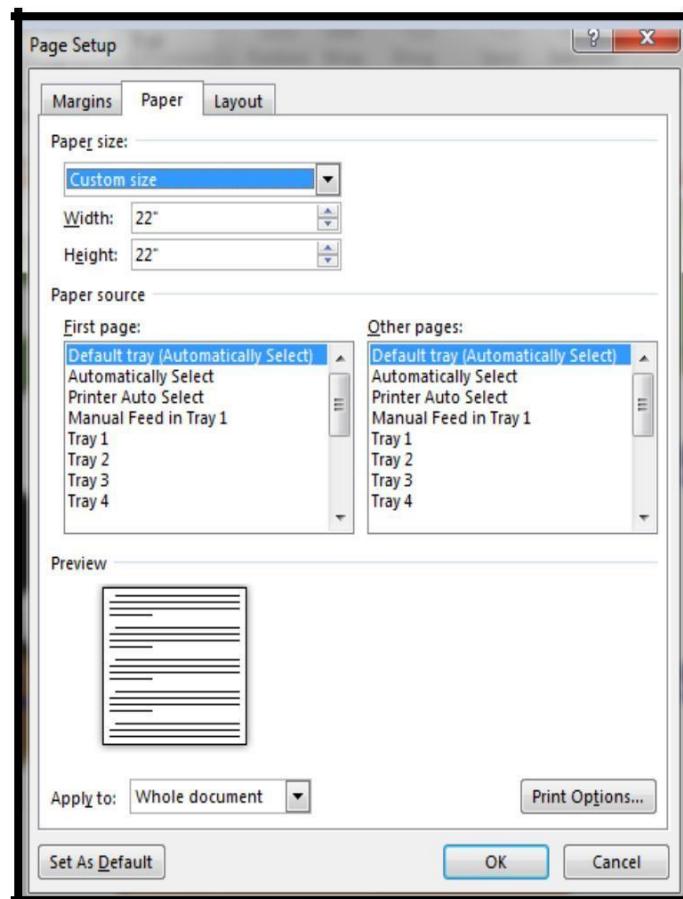


Figure 25: Page setup dialog box

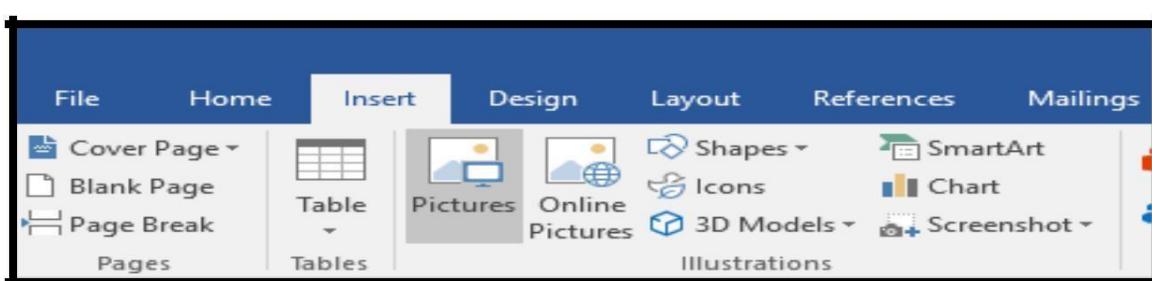


Figure 26: Insert pictures

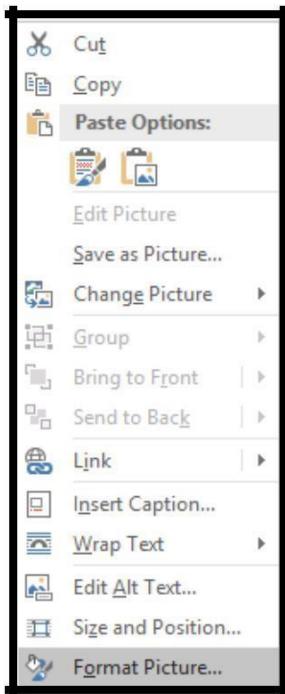


Figure 27: Format picture option

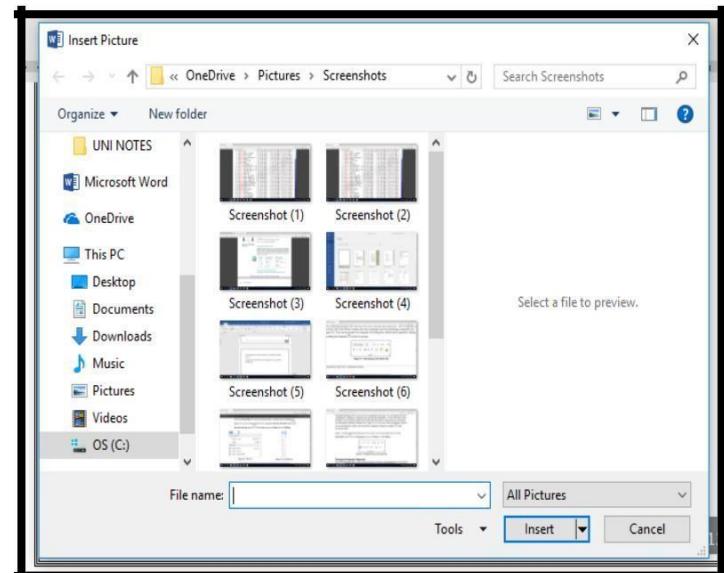


Figure 28: Insert picture dialog box

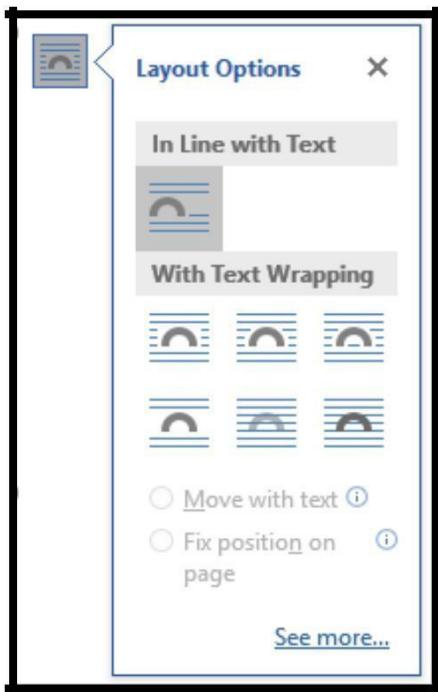


Figure 29: Wrap text options

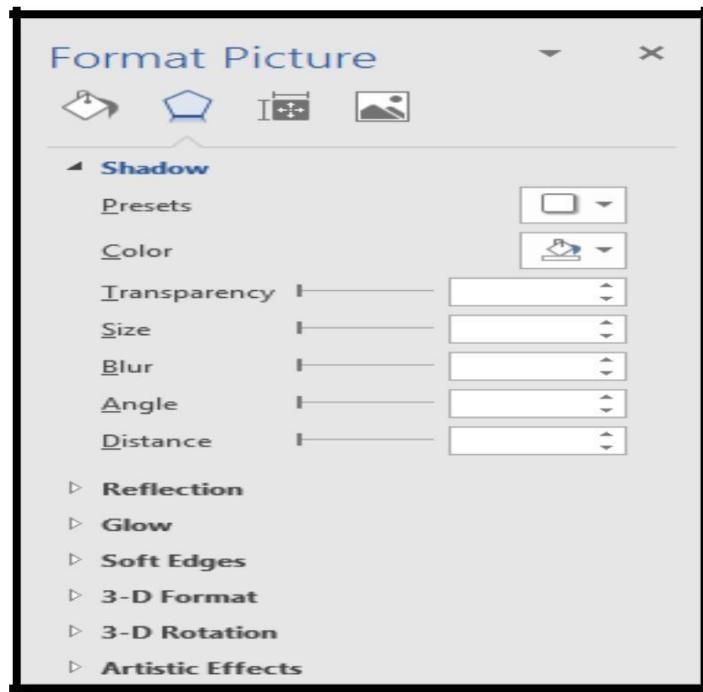


Figure 30: Formatting options



Figure 31: Font color palette

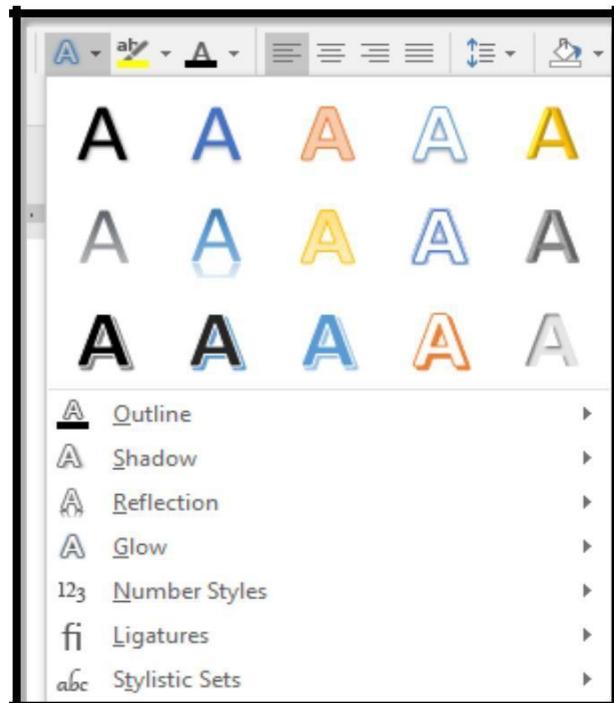


Figure 32: Text effects



Figure 33: Text highlighting colour palette

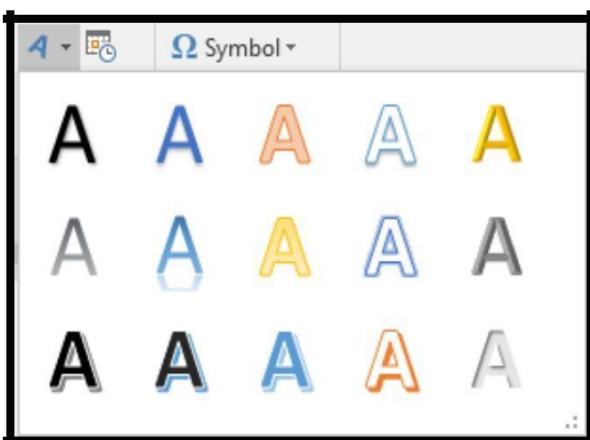


Figure 34: Word art

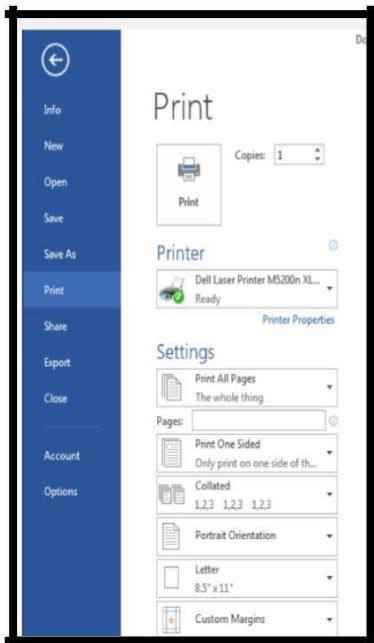


Figure 35: Checking print preview

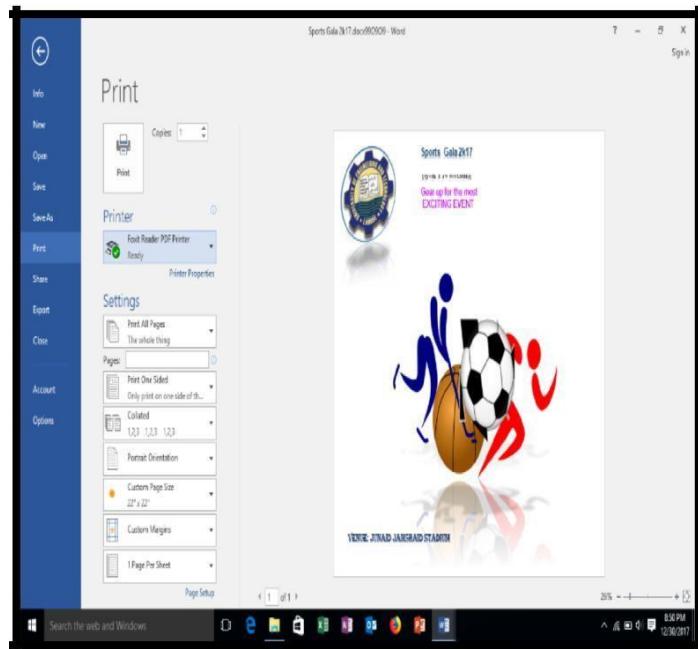


Figure 36: Print preview of document

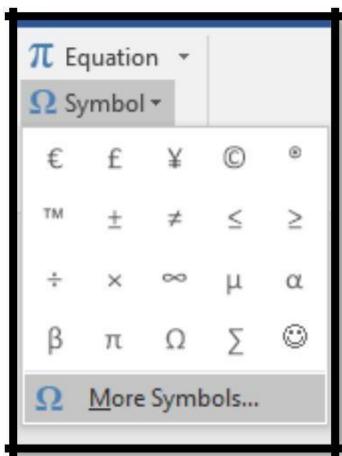


Figure 37: Symbols

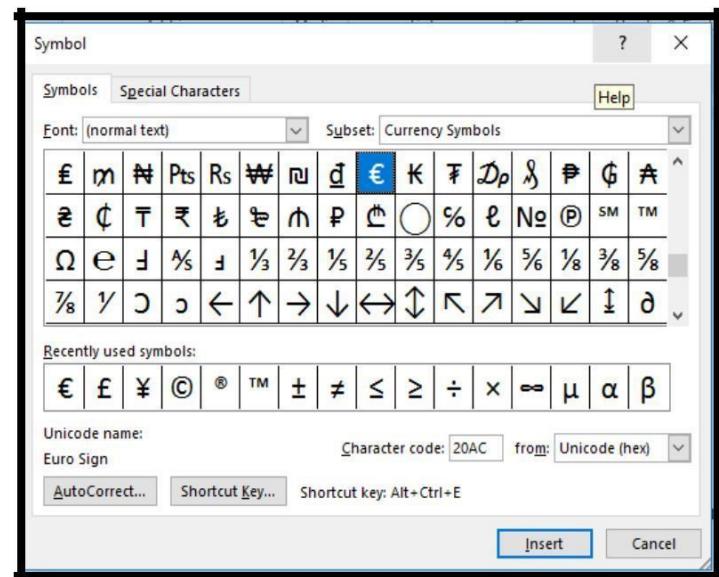


Figure 38: Symbols dialog box

Degree	Year	Board	College/School	Marks

Figure 39: Education information table

### **3.2 CREATE A BIO DATA FORM AND FILL IT.**

1. Open a New blank Word document. Set the paper size to letter and start creating form according to given specimen.
2. Insert picture in the form (UET logo) and select “in line with text” from the wraptext option and add text.
3. Start entering text in the form. Shapes will be inserted in the form. To insert shapes in the form, select the insert tab and in the illustrations group, click on the shapes option and select the desired shape.
4. In this form, we have used rectangle shape to insert picture and symbols and line shape to underline the text.
5. Symbols must be inserted in the form. To do this select the insert tab and in the symbols, group click on the symbol button. (Figure 37)
6. To view all symbols, click on the “more symbols” options. Symbols dialog box appears. And select the tick symbol which is used in the form in this assignment. (Figure 37 and 38)
7. Also, add your education in table. Go to insert tab and click on table. Hover the mouse over the grid and decide how many columns and rows you want. When decided, click on it. Add headings in first row and data in other rows. (Figure 39)

**Student's Information Form**



Session 2017/1<sup>st</sup> Semester  
Department of Computer Science and Engineering  
University of Engineering and Technology, Lahore

1. Name: Mehak Saleem

2. Gender: Male  Female

3. Date of Birth: 16-9-2000 Age: 17 Years 2 Months

4. Residence: Street No # 2, Haffizabad Road, Baghbanpura, Gujranwala.

5. City: Gujranwala  
6. Mobile: 0334-387-9000  
7. Email: saleem.mehak266@gmail.com  
8. Hobbies: Watching Movies  
9. Favorite Movie: A Beautiful Mind  
10. Favorite Book: Secrets

**Father/Guardian's Info**

1. Father/Guardian's Name: Ch. Muhammad Saleem  
2. Father/Guardian's Job Description and Location: Lawyer  
3. Father/Guardian's Contact Number: 0323-609-7602



### **3.3 CREATE REFERENCES IN IEEE FORMAT**

Write a paragraph in Word of 1000 words on any topic you like and read information from at least 5 websites, book, research paper or any document available on internet. Add these 5 references using references tab in word and make sure it appears in the end.

1. On the References tab, in the Citations & Bibliography group, click the arrow next to Style and select IEEE.
2. Then, click on manage sources button. Source manager dialog box will appear. (Figure 40)  
Click on New and the following box will appear. (Figure 41) Select the type of source. Enter all the specifications and click OK.
- 3.
4. Similarly, to create more references again click on new option and select the source type and fill all the specifications.
5. After completing all the references, click on close option.

#### **References**

- [1] "Science abc," [Online]. Available: <https://www.scienceabc.com/innovation/what-is-inside-your-computer-tower-cabinet-cpu-box.html>. [Accessed 2 November 2017].
- [2] "How stuff works," [Online]. Available: <https://computer.howstuffworks.com/inside-computer.htm>. [Accessed 2 November 2017].
- [3] "GCFLearnFree," [Online]. Available: <https://www.gcflearnfree.org/computerbasics/inside-a-computer/1/>. [Accessed 2 November 2017].
- [4] "Computer Hope," [Online]. Available: <https://www.computerhope.com/issues/ch000997.htm>. [Accessed 2 November 2017].
- [5] "ExtremeTech," [Online]. Available: <https://www.extremetech.com/computing/76574-a-beginners-guide-to-what-goes-on-inside-that-pc>. [Accessed 2 November 2017].
- [6] "GCFLearnFree," [Online]. Available: [http://hesperiausd.org/Schools/Middle/HJH/computers/assignments/computer\\_basics/learnfree/comp\\_basics\\_lesson7/7.html](http://hesperiausd.org/Schools/Middle/HJH/computers/assignments/computer_basics/learnfree/comp_basics_lesson7/7.html) [Accessed 2 November 2017].
- [7] "LIFEWIRE," [Online]. Available: <https://www.lifewire.com/tour-inside-a-desktop-pc-2624588>. [Accessed 2 November 2017].
- [8] "Information Vine," [Online]. Available: [https://www.informationvine.com/index?qsref=999&qo=semQuery&ad=semB&o=603146&l=sem&askid=07e992c3-a524-4e92-9a13-b51a628fa0af0-iv\\_gsm&q=inside%20a%20computer&dqi=\\$am=modifiedbroad&an=google\\_s](https://www.informationvine.com/index?qsref=999&qo=semQuery&ad=semB&o=603146&l=sem&askid=07e992c3-a524-4e92-9a13-b51a628fa0af0-iv_gsm&q=inside%20a%20computer&dqi=$am=modifiedbroad&an=google_s). [Accessed 2 November 2017].
- [9] "NetLiteracy," [Online]. Available: [http://www.netliteracy.org/wp-content/uploads/2012/07/Basic-Computer-Skills1.pdf?gid=EAraQobChMIG9u5mI-51wIVjRwrCh1UcwECEAAAYyAEgJ36ID\\_BwE](http://www.netliteracy.org/wp-content/uploads/2012/07/Basic-Computer-Skills1.pdf?gid=EAraQobChMIG9u5mI-51wIVjRwrCh1UcwECEAAAYyAEgJ36ID_BwE). [Accessed 2 November 2017].
- [10] "Computers4Africa," [Online]. Available: <http://computers4africa.org.uk/get/>. [Accessed 3 November 2017].
- [11] "comptech," [Online]. Available: <http://www.comptechdoc.org/hardware/pc/begin/hwininside.html>. [Accessed 3 November 2017].
- [12] "FixYourComputer," [Online]. Available: <http://www.fix-your-computer-today.com/inside-the-computer.html>. [Accessed 3 November 2017].
- [13] [Online]. Available: <http://openbookproject.net/courses/intro2ict/hardware/internal.html>. [Accessed 3 November 2017].
- [14] "The BAM Edition," [Online]. Available: <http://sites.jmu.edu/103oconnor-16/introduction-to-basic-computer-components-and-functions/>. [Accessed 3 November 2017].
- [15] "MUO," [Online]. Available: <http://www.makeuseof.com/tag/whats-inside-your-computer-the-story-of-every-component-you-need-to-know-3/>. [Accessed 3 November 2017].

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

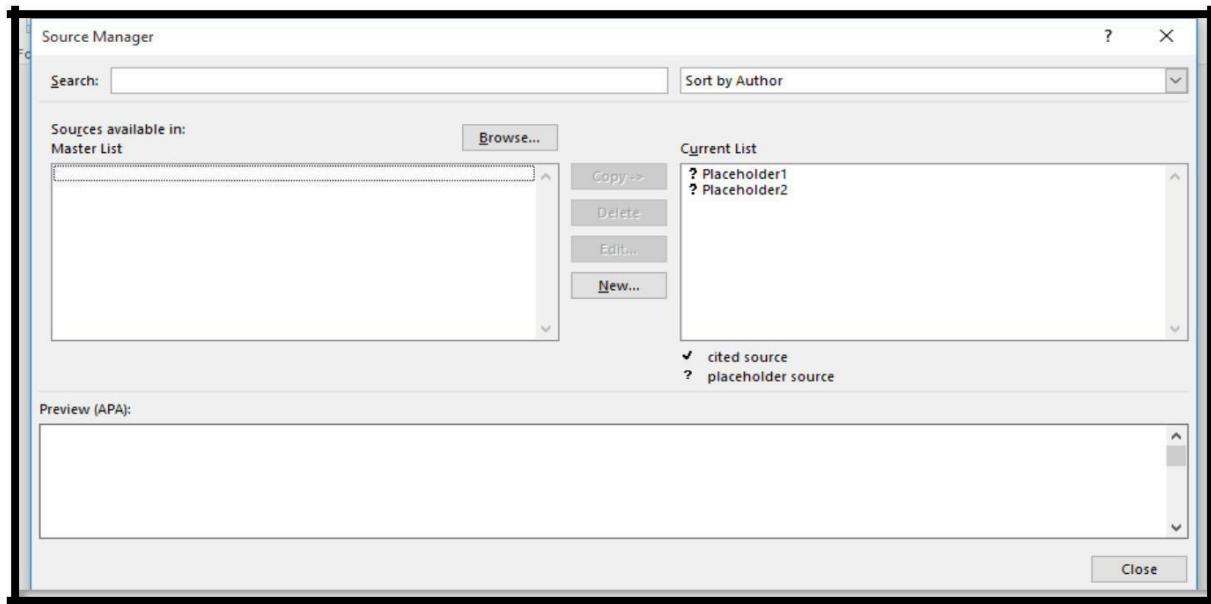


Figure 40: Manage resources

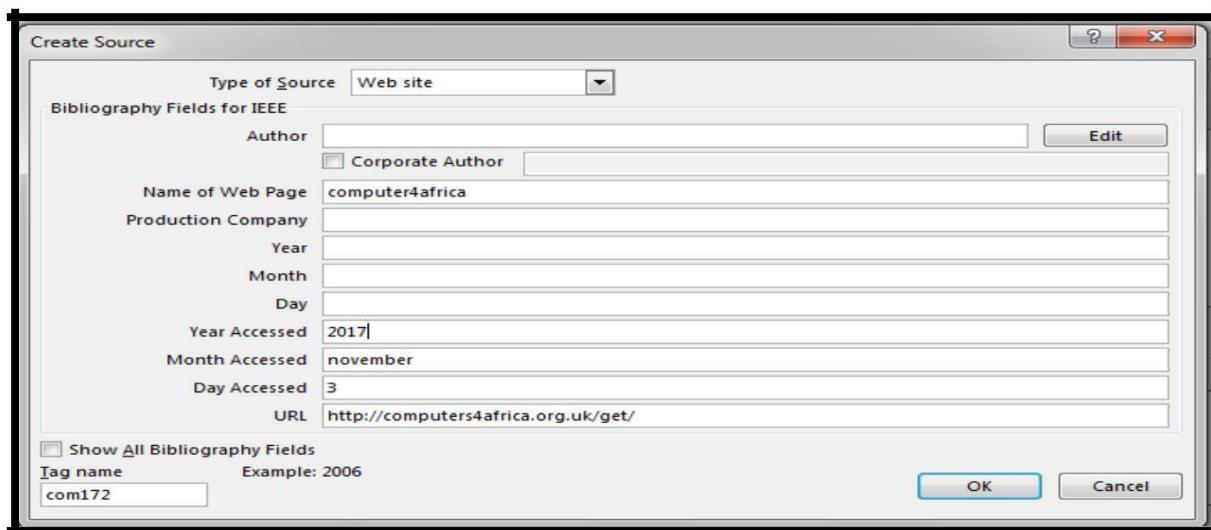


Figure 41: Create source dialog box



Figure 42: Bibliography

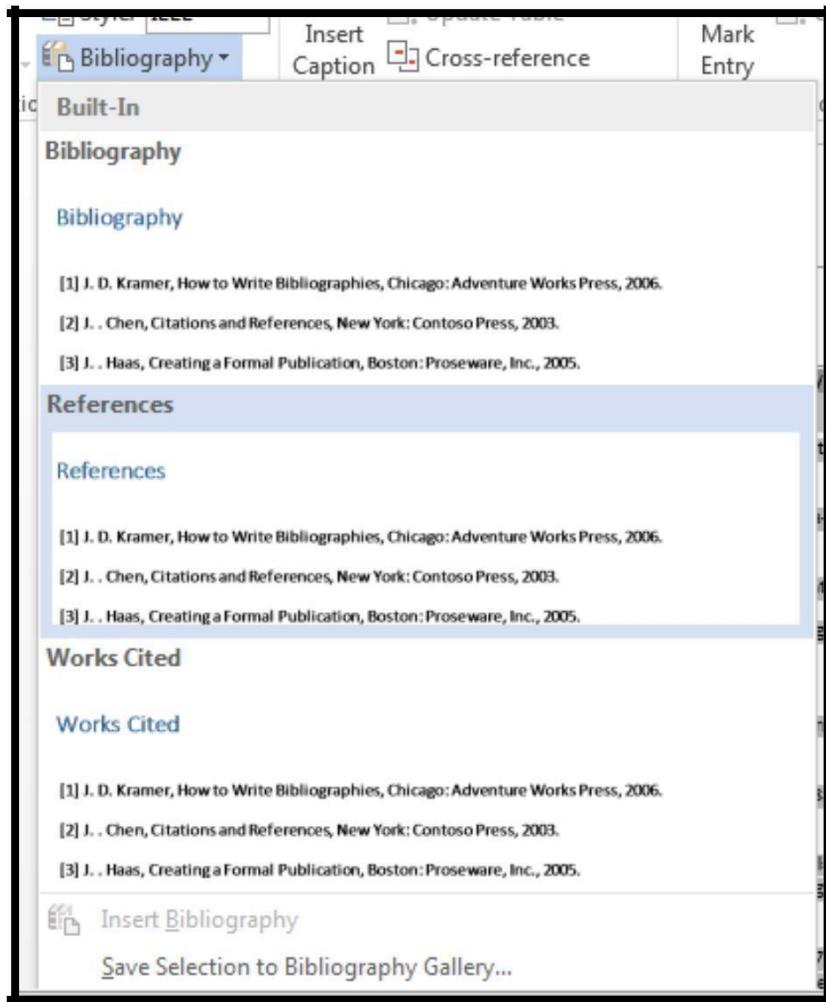


Figure 43: Design option for references

---

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO1</b>	Use state of art tools to develop professional documents, spread sheets, web layouts and presentations in various assignments CLO Description/Text					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
<b>Total</b>						<b>/10</b>

## **1 LAB 3 (INTRODUCTION TO ANALYSIS TOOL)**

---

### **1.1 GOALS OF THE DAY:**

- Introduction to Microsoft Excel

## **2 SECTION A**

---

Microsoft Excel is a powerful electronic spreadsheet program you can use to automate accounting work, organize data, and perform a wide variety of tasks. Excel is designed to perform calculations, analyze information, and visualize data in a spreadsheet. Also, this application includes database and charting features.

When you create a new Excel 2016 workbook, the app presents a blank workbook that contains one worksheet. You can add or delete worksheets, hide worksheets within the workbook without deleting them, and change the order of your worksheets within the workbook. You can also copy a worksheet to another workbook or move the worksheet without leaving a copy of the worksheet in the first workbook. If you and your colleagues work with many documents, you can define property values to make your workbooks easier to find when you and your colleagues attempt to locate them by using the Windows search box.

Another way to make Excel easier to use is by customizing the Excel app window to fit your work style. If you find that you use a command frequently, you can add it to the Quick Access Toolbar so it's never more than one click away. If you use a set of commands frequently, you can create a custom ribbon tab, so they appear in one place. You can also hide, display, or change the order of the tabs on the ribbon.

To save your workbook by using a keyboard shortcut, press Ctrl+S. Students often ask, "**How often should I save my files?**" It is good practice to save your changes every half hour or even every five minutes, but the best time to save a file is whenever you make a change that you would hate to have to make again.

Open Microsoft Excel and create a file named Lab2.

### **2.1 TABS IN EXCEL:**

There are following tabs in excel.

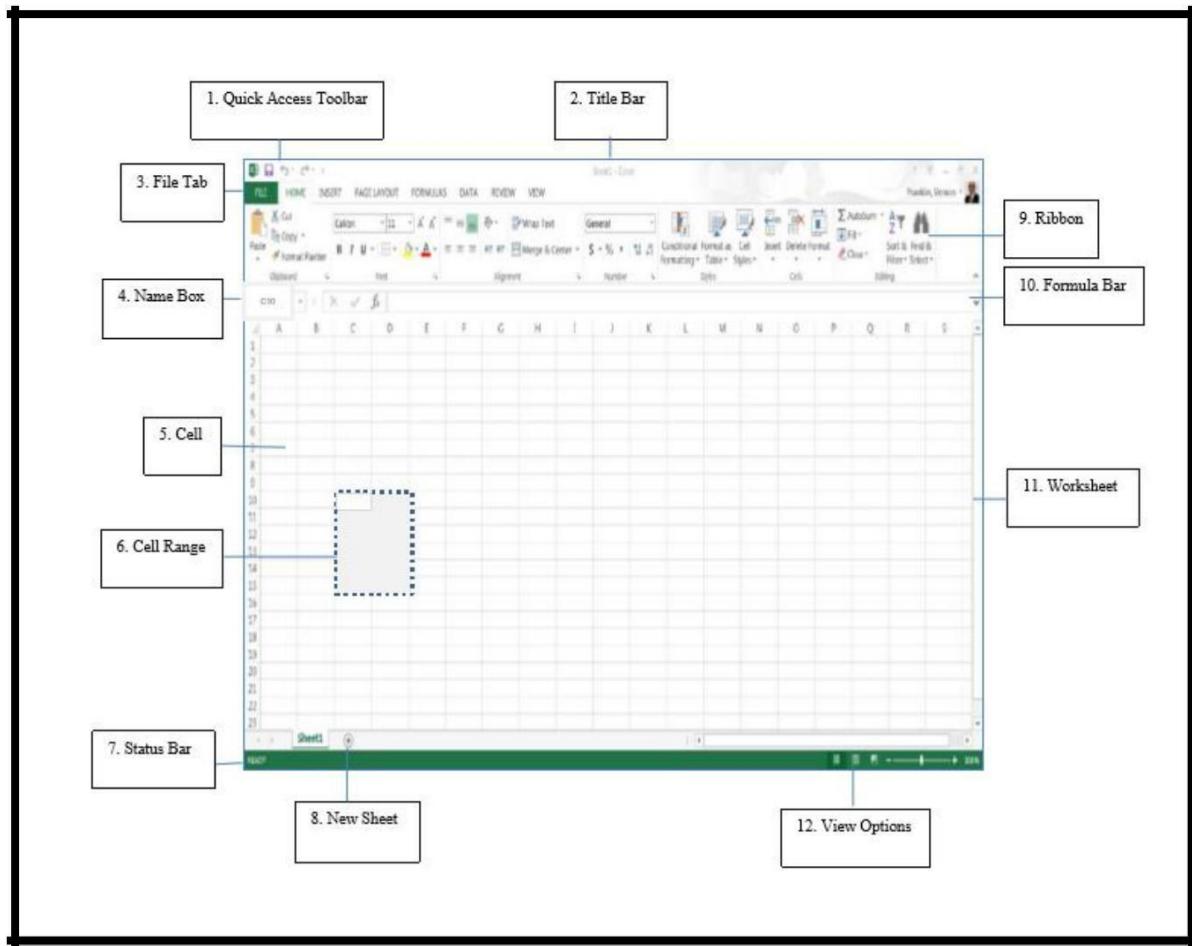
**Home tab:** it has many important groups like font, alignment, numbers, cells group which has many important features like font size, font style, merge cells etc.

**Insert tab:** it has many important features like inserting pictures, tables, charts, header and footer.

**Formulas tab:** used to insert formulas in sheet. It's insert function and AutoSum features are mostly used.

**Review tab:** its spelling and thesaurus features are mostly used.

## 2.2 FEATURES OF EXCEL



**Quick Access Toolbar:** Displays quick access to commonly used commands.

**Title Bar:** Displays the name of the application file.

**File Tab:** The File tab has replaced the Office button. It helps you to manage the Microsoft application and provide access to its options such as Open, New, Save, As Print, etc.

**Name Box:** Displays the active cell location

**Cell:** The intersection of a row and column; cells are always named with the column letter followed by the row number (e.g. A1 and AB209); cells may contain text, numbers and formulas. 6

**Range:** One or more adjacent cells. A range is identified by its first and last cell address, separated by a colon. Example ranges are B5:B8, A1:B1 and A1:G240.

**Status Bar:** Displays information about the current worksheet.

**New Sheet:** Add a new sheet button

**Ribbon:** Displays groups of related commands within tabs. Each tab provides buttons for commands.

**Formula Bar:** Input formulas and perform calculations

**Worksheet:** A grid of cells that are more than 16,000 columns wide (A-Z, AA-AZ, BA-BZ...IV) and more than 1,000,000 rows long.

**View Option:** Display worksheet view mode.

<b>Method</b>	<b>Description</b>
<b>mouse pointer</b>	Use the mouse pointer  to select a cell.
<b>scroll bars</b>	Use the horizontal and vertical scroll bars to move around the spreadsheet to view columns and rows not currently visible. Click the mouse pointer once the desired cell is visible.
<b>arrow keys</b>	Use the left $\leftarrow$ , right $\rightarrow$ , up $\uparrow$ , and down $\downarrow$ arrows to move accordingly among cells.
<b>Enter</b>	Press the Enter key to move down one cell at a time.
<b>Tab</b>	Press the Tab key to move one cell to the right.
<b>Ctrl+Home</b>	Moves the cursor to cell A1.
<b>Ctrl+End</b>	Moves the cursor to the last cell of used space on the worksheet, which is the cell at the intersection of the right-most used column and the bottom-most used row (in the lower-right corner).
<b>End + arrow key</b>	Moves the cursor to the next or last cell in the current column or row which contains information.

*Figure 1: Methods to navigate around a sheet*

**Formula:** You can create any type of math calculation on your own using the following mathematical operators:

<b>Symbol</b>	<b>Meaning</b>
=	equals - used to begin a calculation
+	addition
-	subtraction
*	multiplication
/	division
^	exponentiation
(	open parenthesis - used to begin a grouping
)	close parenthesis - used to close a grouping

*Figure 2: Symbols used in formulas*

For example: = A1 + A5 ....

Let's start practicing. Add some numbers in cells of your choice as shown in Figure 3.

A	B	C	D	E
Roll numbers	Physics marks	Math marks	Chemistry marks	Total
1	23	30	37	
2	45	38	55	
3	65	55	60	
4	33	25	20	
5	76	80	85	

Figure 3: Dummy data written in excel worksheet

1. Click in the empty cell which will contain the formula. In current example case, it will be E2.
2. Type an equal sign (=). (Figure 4)
3. Type the cell address or click the cell that contains the first number. (Figure 4)
4. Type the math operator (+ - \* ^). (Figure 4)
5. Type the cell address or click the cell that contains the second number. (Figure 4)
6. Continue in this manner until the formula is complete.
7. Use parenthesis for clarification.
8. Press the Enter key. Result will be added to that cell.
9. To use same formula for all rows, hover mouse to the right most bottom point until mouse pointer changes to plus sign (+). Click at corner of cell and drag it to the bottom of the same column. Formula will automatically be copied to other cells according to their cell references.

D2	A	B	C	D	E
	Roll numbers	Physics marks	Math marks	Chemistry marks	Total
1					
2	1	23	30	37	=B2+C2+D2
3	2	45	38	55	
4	3	65	55	60	
5	4	33	25	20	
6	5	76	80	85	

Figure 4: Adding sum formula for addition of marks

### 2.3 INSERT FUNCTIONS:

Go to formula tab. This selection demonstrates how to use the Insert Function menu to create a formula. Click on the Insert Function button or from the AutoSum drop-down arrow and select More Functions to display a list of over 200 functions available in Excel. The Insert Function dialog box displays the

function categories from the drop-down menu list. The function names will appear in the function name box below. (Figure 5)

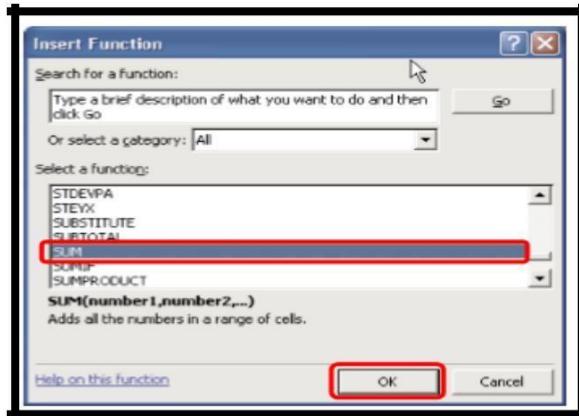


Figure 5: Insert function dialog box

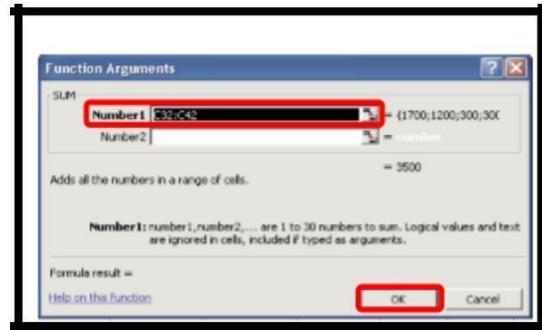


Figure 6: Function argument box

Once you select a category and a function name, click on the OK button. The Function Arguments palette will appear. (Figure 6) Type any numbers, cell addresses, ranges, or any other parameters in the required boxes, and then click on the OK button to insert the completed formula in the spreadsheet.

### AutoSum:

Adding is the most common math operation performed in Excel. The Home ribbon includes an AutoSum button for adding. This button provides a shortcut to typing formulas. (Figure 7)

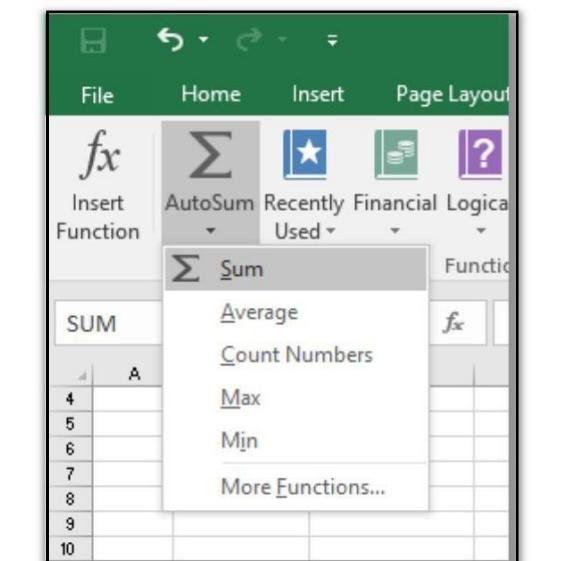


Figure 7: AutoSum button

### Maximum (MAX):

Maximum indicates the largest value in the designated list of numbers.

Syntax =MAX (number1, number2...)

Examples =MAX (A15:A35)  
=MAX (D10:D200, D225:D325)

### Sum:

Adding is the most common function performed in Excel. The SUM function adds values. Specify values, individual cell addresses and/or range addresses in the number's variables.

Syntax =SUM(number1,number2,...)

Examples =SUM(A10:A25)  
=SUM(B15:C20)

### Average:

An average sum all values and divides by the total number of values. Specify values, individual cell addresses and/or range addresses in the number's variables. You can simply select the range of cells and select average from AutoSum button

Syntax =AVERAGE (number1, number2...)

Examples =AVERAGE (15, 255, 45)  
=AVERAGE (B2:B18)

### Minimum (MIN):

Minimum indicates the smallest value in the designated list of numbers.

Syntax =MIN (number1, number2...)

Examples =MIN (A15:A35)  
=MIN (D10:D200, D225:D325)

### **3 LAB TASKS (TO BE PERFORMED IN LAB)**

---

#### **3.1 CREATE ACADEMIC RECORD OF A STUDENT.**

##### **Steps:**

1. Start Microsoft Excel and open blank workbook. To do this, select the new tab and click on blank workbook.
2. Start typing the given record in the workbook. In this case, I have given heading at the top and then inserted picture. To insert picture, go to the insert tab and select insert picture button. The insert picture dialog box will appear. Figure 8
3. Select the desired picture and click on insert button at the bottom of dialog box.
4. Enter the given data in the workbook. To underline the text, I have used shapes. Go to the insert tab and click the down arrow to the right of shapes button. Figure 9
5. Use the line shape to underline the text
6. To change the appearance of the text, go to Home tab and in the font group, font style and sizes etc. can be changed. And in the Alignment group, Merge & Centre is used to merge the text. Figure 10
7. To calculate total of the values, place the cursor in the cell where you want to display total and go to the Formula tab and click the down arrow of AutoSum and select Sum. Figure 11
8. And select the values you want to total as follows: Figure 12.
9. Similarly, calculate average of all values. Go to the formulas tab and in the AutoSum button select average and then select the cell on which you want to apply average formula. For example, in Figure 13.
10. To add colour shading to values, select the values and in the home tab in the font group select the desired colour from fill colour
11. To calculate percentage, I have typed the following formula in the formula bar according to my data. Figure 15: Percentage calculationFigure 15
12. After calculations, you can save record. Go to the file tab and select saves as. Then select browse.
13. Save as dialog box will appear. Enter the name of the file and location where you want to save the file. Then click save button. File will be saved.

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

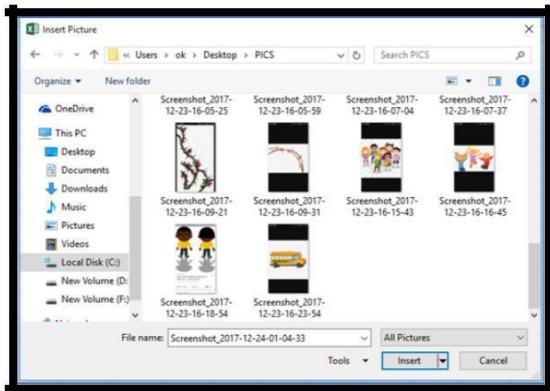


Figure 8: Insert picture dialog box



Figure 9: Shapes

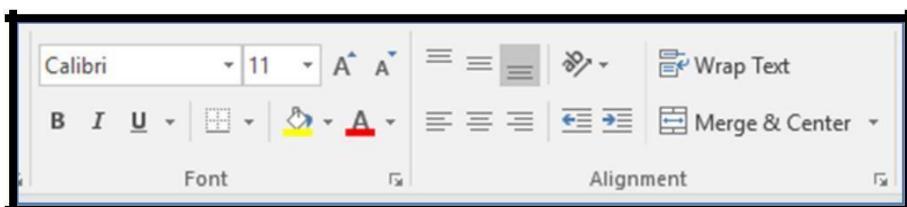


Figure 10: Font and alignment group

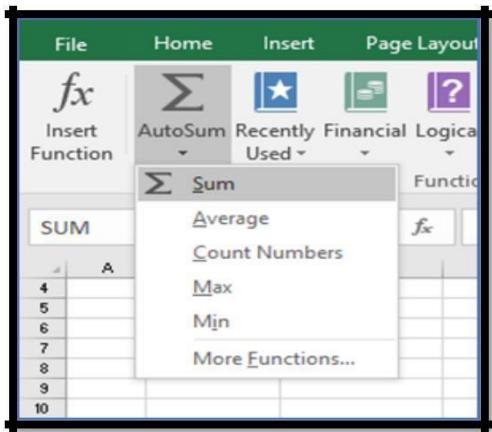


Figure 11: Calculation sum

	A	B
1	0	
2	5	
3	3	
4	4	
5	2	
6	8	
7	6	
8	5	
9	4	
10	4	
11	41	
12	82	
13	$=SUM(A1:A12)$	

Figure 12: Sum calculation

	A	B
1	0	
2	5	
3	3	
4	4	
5	2	
6	8	
7	6	
8	5	
9	4	
10	4	
11	41	
12	82	
13	164	
14	164	
15	$=AVERAGE(A1:A14)$	

Figure 13: Calculating average

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

ST NO. 2 HAFIZABAD ROAD, GUJRANWALA					
NAME OF INSTITUTION	CITY	CLASS	TOTAL MARKS	OBTAINED MARKS	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	1	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	2	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	3	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	4	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	5	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	6	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	7	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	8	800	700	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	9	550	475	
ST DENY'S ENGLISH SCHOOL	GUJRANWALA	10	550	521	
PUNJAB COLLEGE	GUJRANWALA	11	550	520	
PUNJAB COLLEGE	GUJRANWALA	12	550	520	
		TOTAL	8600	7636	
		AVERAGE		636.3333333	
		PERCENTAGE		88.79069767	

Figure 15: Percentage calculation

Student's Academic Record					
 <b>Department Of Computer</b> <b>Science and Engineering</b> <b>UET, Lahore</b>					
<b>Name of Stud:</b> MEHAK SALEEM <b>Name of Fatk:</b> CH. MUHAMMAD SALEEM <b>Residence:</b> ST NO. 2 HAFIZABAD ROAD, GUJRANWALA					
SR No.	NAME OF INSTITUTION	CITY	CLASS	TOTAL MARKS	OBTAINED MARKS
1	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	1	800	800
2	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	2	800	780
3	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	3	800	780
4	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	4	800	780
5	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	5	800	780
6	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	6	800	780
7	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	7	800	780
8	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	8	800	780
9	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	9	550	475
10	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	10	550	521
11	PUNJAB COLLEGE	GUJRANWALA	11	550	520
12	PUNJAB COLLEGE	GUJRANWALA	12	550	520
		TOTAL	8600	7730	
		AVERAGE		644.66666667	
		PERCENTAGE		88.55548832	

Figure 14: Sample output

### 3.2 CREATE PIE AND BAR CHART WITH THE DATA OF STUDENT'S RECORD PROVIDED IN THE PREVIOUS ASSIGNMENT.

1. First, select the data values of which you want to create the chart. In case of pie chart, it is between obtained and total marks. Select the values. See Figure 18
2. Then go to the insert tab and select pie chart in chart group. And select 2-D in pie chart option. See Figure 19 for details.
3. The pie chart will be created according to the data values. (Figure 20)
4. Now click on the chart title text box to enter the title and to change colour scheme. (Figure 21)
5. After adding title and changing colours, save the chart with specified name.
6. For bar chart, select the required data values as shown in Figure 22.
7. To create chart, go to the insert tab and in the chart group, select the recommended chart and select the clustered column chart. (Figure 23)
8. The chart will be created. Add title of the chart and tick the check boxes of which chart elements you want to include in your chart. (Figure 24)
9. In this chart five chart elements are included, which are shown in the above figure. Legends are defined at the bottom of graph. (Figure 25)
10. Add axis titles and chart titles and you can also change the colour and style of graph.
11. After doing all above steps, chart will be created in desired form and it can now be saved.
12. See Figure 16 and Figure 17 for sample output.

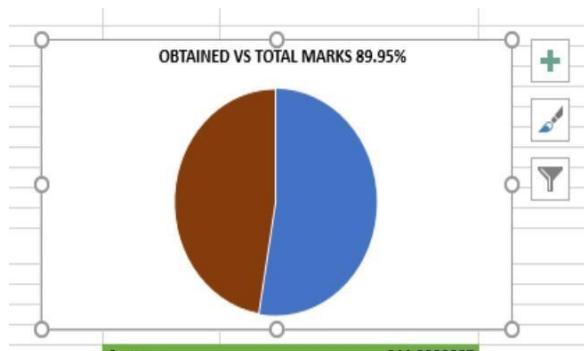


Figure 16: Output Pie chart

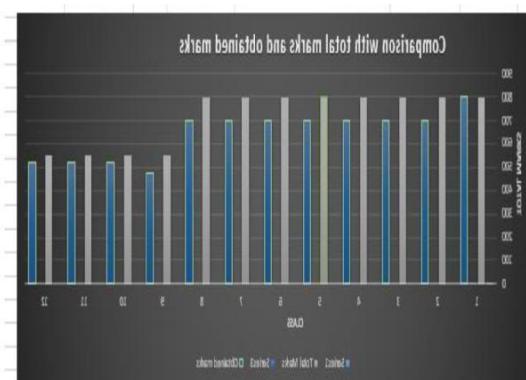


Figure 17: Output Bar chart

SR NO.	NAME OF INSTITUTION	CITY	CLASS	TOTAL MARKS	OBTAINED MARKS
1	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	1	800	800
2	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	2	800	700
3	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	3	800	700
4	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	4	800	700
5	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	5	800	700
6	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	6	800	700
7	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	7	800	700
8	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	8	800	700
9	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	9	550	475
10	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	10	550	521
11	PUNJAB COLLEGE	GUJRANWALA	11	550	520
12	PUNJAB COLLEGE	GUJRANWALA	12	550	520
TOTAL				8600	7736
AVERAGE					644.6666667
PERCENTAGE					89.95348837

Figure 18: Selecting values for graph

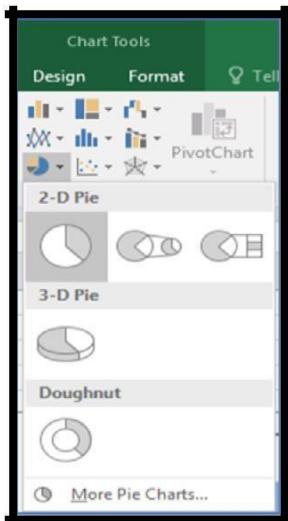


Figure 19: 2D Pie Chart

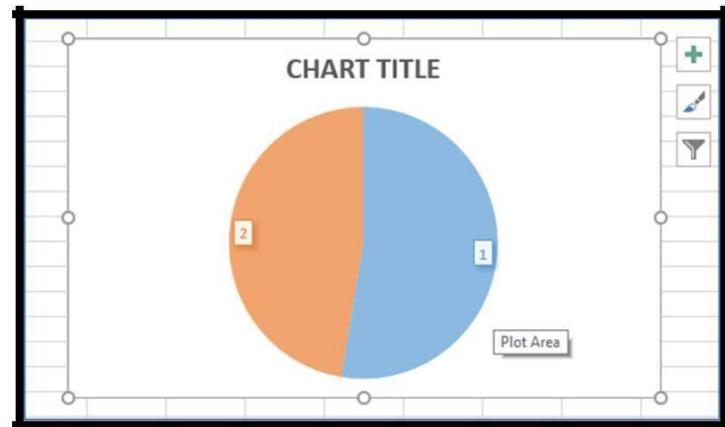


Figure 20: Pie chart

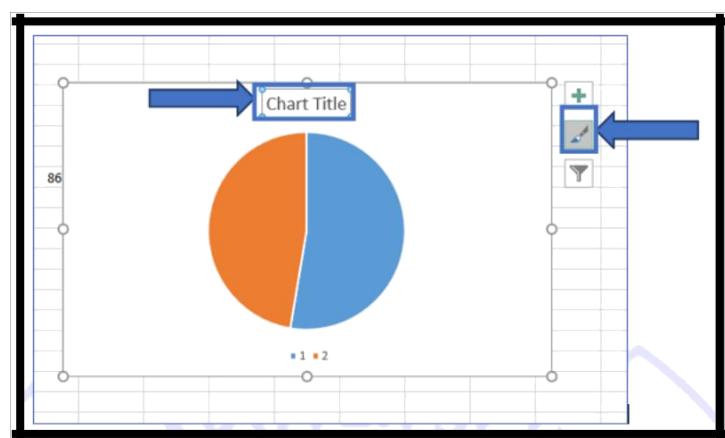


Figure 21: Adding chart title and colours

SR NO.	NAME OF INSTITUTION	CITY	CLASS	TOTAL MARKS	OBTAINED MARKS
1	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	1	800	800
2	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	2	800	700
3	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	3	800	700
4	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	4	800	700
5	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	5	800	700
6	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	6	800	700
7	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	7	800	700
8	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	8	800	700
9	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	9	550	475
10	ST DENY'S ENGLISH SCHOOL	GUJRANWALA	10	550	521
11	PUNJAB COLLEGE	GUJRANWALA	11	550	520
12	PUNJAB COLLEGE	GUJRANWALA	12	550	520
				<b>TOTAL</b>	<b>8600</b>
				<b>AVERAGE</b>	<b>644.66666667</b>
				<b>PERCENTAGE</b>	<b>89.95348837</b>

Figure 22: Selecting values for bar chart

# University Of Engineering and Technology, Lahore

## Computer Engineering Department

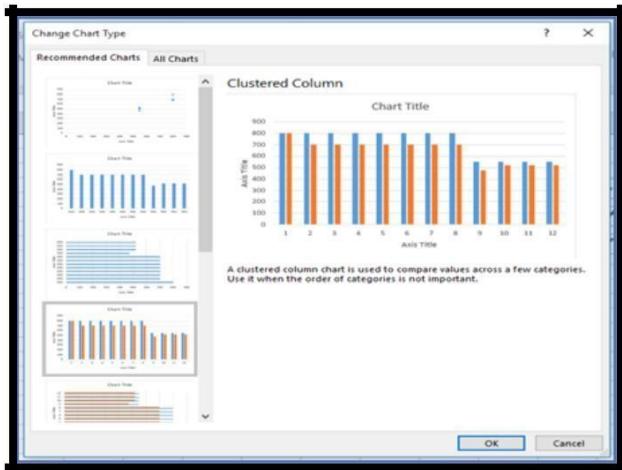


Figure 23: Creating bar chart

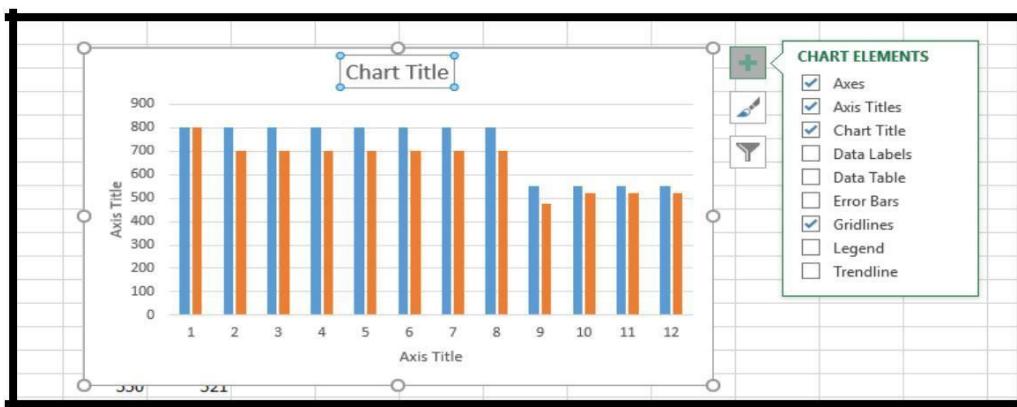


Figure 24: Chart elements

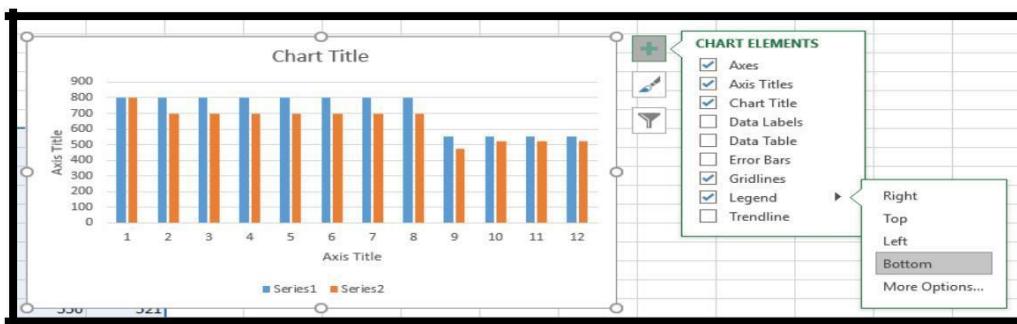


Figure 25: Adding legends

### 3.3 CREATE A PLAN CHART AS GIVEN IN SAMPLE.

1. Open a new blank Excel document.
2. Select the cells you want to merge. On **Home** tab, in **Alignment** group, click on **Merge and centre**. (Figure 26)
3. Add text in it and change its font style, colour and size. (Figure 28)
4. On **Home** tab, in **Font** group, click on arrow 
5. From the drop down, menu select **Line Style** and then select line of desired thickness. (Figure 28) 
6. Again, click on arrow , click on **Line Colour** and select desired colour for border. (Figure 29)
7. A pencil will appear. Draw the border the cells.
8. Select the cells whose column width you want to change. On **Home** tab, in **Cells** group, click on arrow below **Format**  and select **Column Width** from drop down menu. (Figure 30)
9. Change the **Column Width** and click OK. (Figure 35)
10. Column width of selected cells will be changed.
11. Merge the required cells by the methods described previously.
12. To fill colour in cells, right click on it and select **format cells** from the dropdown menu. (Figure 31)
13. Click on **Fill** from **Format Cell** dialogue box and select desired fill colour. (Figure 32) 
14. Select the merged cells and on Home tab, in Alignment group, click on  and then select Vertical Text from drop down menu. (Figure 34)
15. Add text. It will appear vertically. (Figure 36) 
16. Again, select other merged cell, click on  and select **Rotate Text Up** from the list.
17. Add text in cell and text will appear in this way.
18. In some cells, text will appear in this way. (Figure 37)
19. To resolve this, select the cell and click on **Home** tab, in **Alignment** group, select **Wrap Text**. (Figure 40)
20. Text in cell will then appear in the form of Figure 38.
21. Text in the cell is **Upper Aligned**. (Figure 39)
22. Complete the task and save the document.
23. See Figure 33 for sample output.

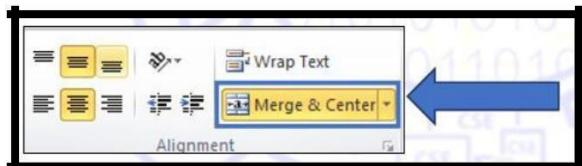


Figure 26: Cells are merging



Figure 27: Text is formatted

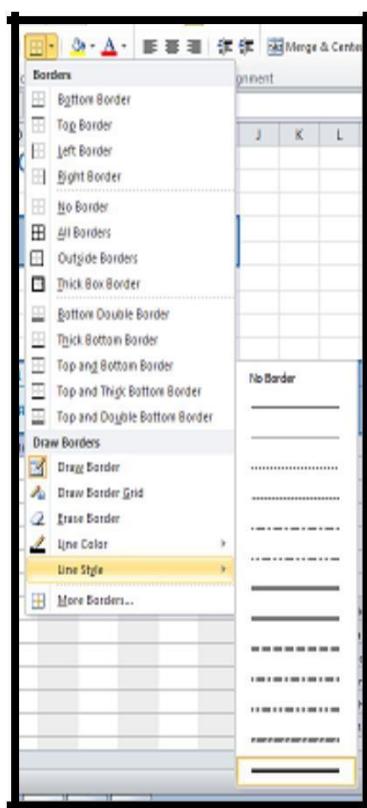
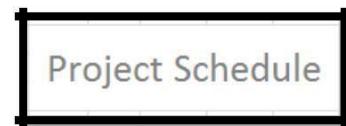


Figure 28: Adding borders to cells

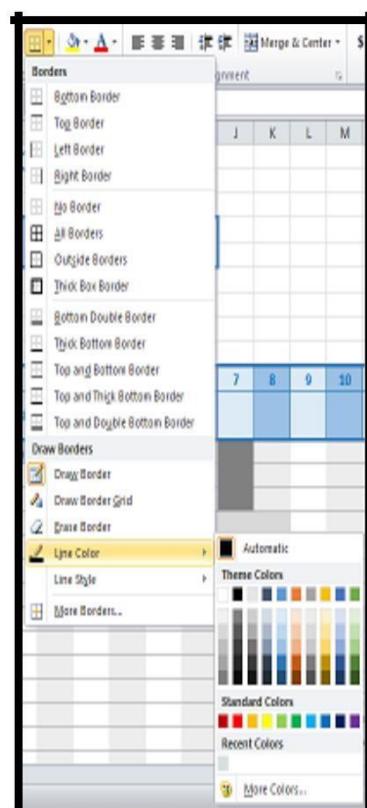


Figure 29: Changing colour of border

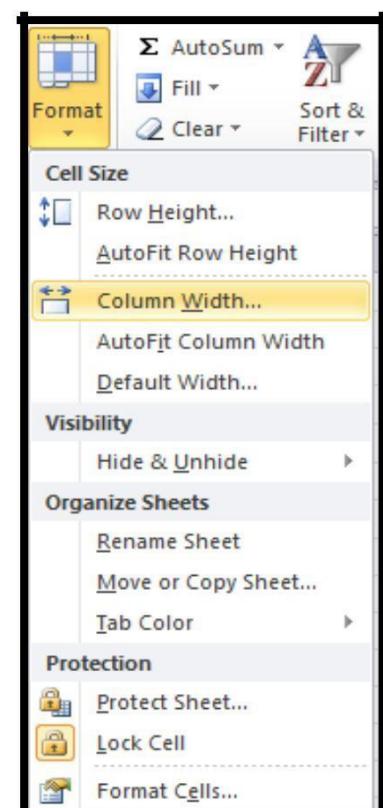


Figure 30: Changing of column width

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

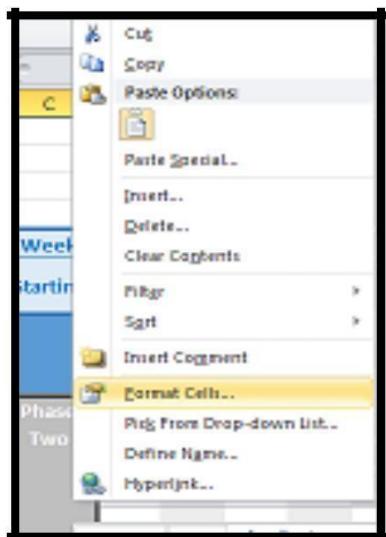


Figure 31: Formatting of cells

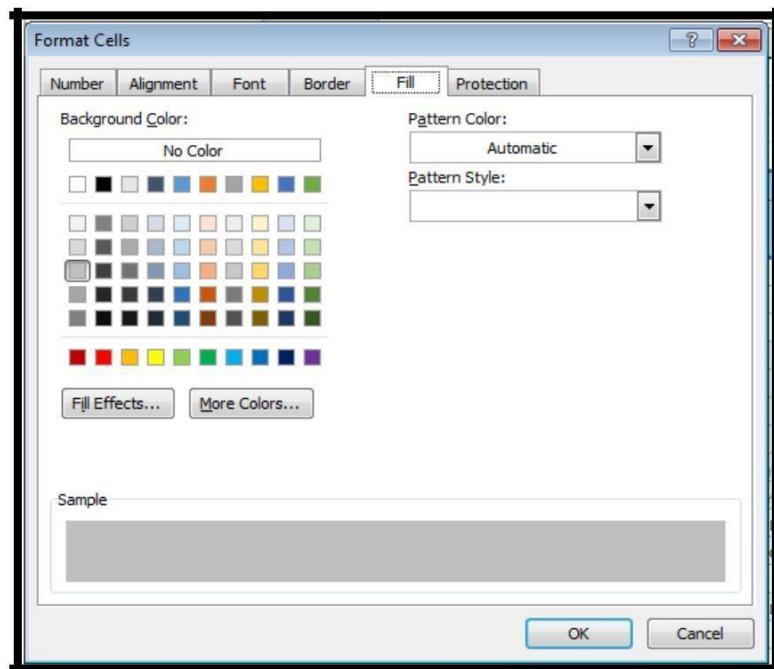


Figure 32: Colour fill of cells is changing

Day	1	2	3	4	5	6	7	8	
Prefetching	Department Selected								
	collected teachers' info								
	submitted report								
Phase Two									
Survey				Conducted survey					
Designing and filling of google form					Google form was designed				
Survey Analysis						google form was filled			
Presentation							Data was presented		
Report									

Figure 33: Sample output

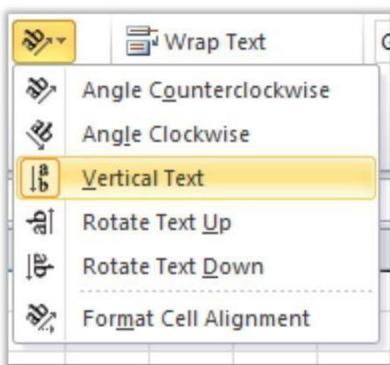


Figure 34: Insertion of vertical

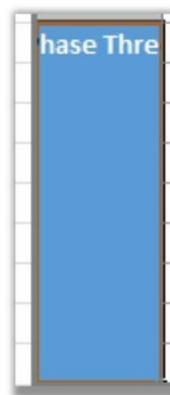
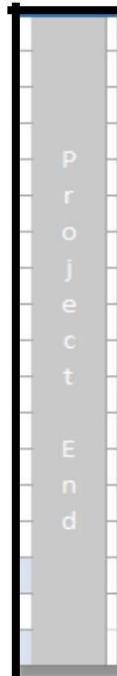


Figure 37: Text in cell without wrapping

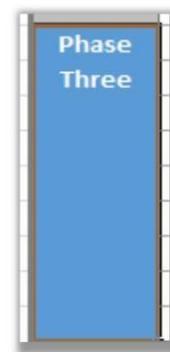


Figure 38: Text is wrapped in cell

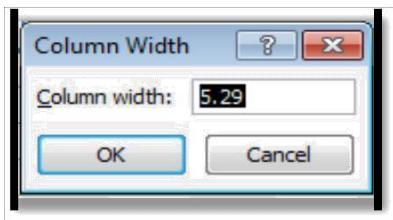


Figure 35: Writing a column

Figure 36: Vertical text is shown



Figure 39: Alignment buttons are shown

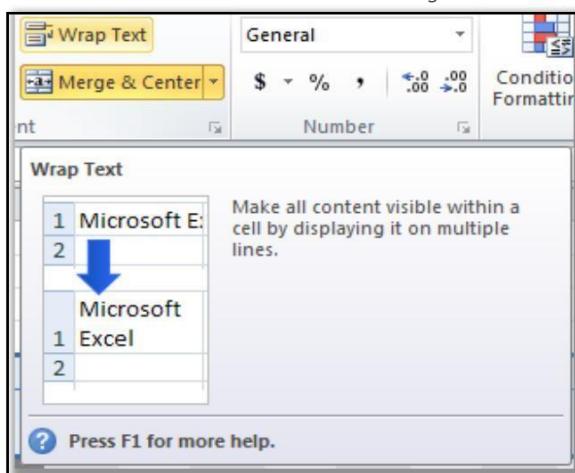


Figure 40: Wrapping of text

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO2</b>	Understand general problem-solving strategies in development of computer algorithms.					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
					<b>Total</b>	<b>/10</b>

## **Lab No. 6**

### **Objective:**

- To understand the basics of Computer System Programming.
- To compose and run a simple program from scratch.

### **Pre-Lab Reading:**

Program is a set of instructions used to solve a particular problem. A computer is a dumb machine. It cannot do anything on its own. It can only act on the given instructions.

### **Programming:**

Computer can only understand a sequence of 1s and 0s. The following looks strange to use but, in fact is how computer reads and executes everything that it does:

101010001110010101010111100010101110

This is called Machine language. We can imagine how complicated programming would be if we had to learn this complex language. That in fact, was how programming was done many years ago; however today we are fortunate to have what are called high level languages such as C, java and C++, python etc. These languages are more geared for human understanding and thus make programming much easier. However, computer only understands machine language, there must be a translation process to convert these high-level languages to machine code. This is often done by a **compiler**.

Compiler analyzes a program developed in a particular computer (high level) language and then translates it into a form (machine language) that is suitable for execution on your particular computer system. The program that is to be compiled is first typed into a file on the computer system.

In the first step of the compilation process, the compiler examines each program statement contained in the source program and check it to ensure that it conforms to the syntax and semantics of the language. If any mistakes are discovered by the compiler during this phase, they are reported to the user and the compilation process ends right

there. The errors then have to be corrected in the source program (with the use of an editor) and the compilation process must be restarted. Typical errors reported during this phase of compilation might be due to an expression that has unbalanced parentheses (syntactic error), or due to the use of a variable that is not “defined” (semantic error).

Once we have executable code, the program is ready to be run. Hopefully it will run and everything will be fine, however that is not always the case. During “run time”, we may encounter another kind of error called a **run time error**. This error occurs when we ask computer to do something it cannot do. The computer cannot violate the laws of mathematics and other binding restrictions.

Asking the computer to divide a number by zero is an example of a run time error. We get executable code; however, when the program tries to execute the command to divide by zero, the program will stop with a run time error. Run time errors, are usually more challenging to find than syntax errors. Once we run our program and get neither syntax nor run time errors, are we free to rejoice? Not exactly. Unfortunately, we may still encounter the worst type of error: the dreaded Logic error. Whenever we ask the computer to do something, but mean for it to do something else, we have a logic error. There must be precise and clear instructions that generate our intentions to the computer. The computer only does what we ask it to do.

Asking it to multiply by 3 when we want something doubled is an example of a **logic error**. This type of error is difficult to identify as compiler cannot detect these errors, the only possible way is to check that the output/result of your program matches with your expectations or not.

### **Python:**

Python is a general-purpose programming language that can be used effectively to build almost any kind of program that does not need direct access to the computer’s hardware. Python is a highlevel, dynamic programming language. The language was first created in the early 1990s, and reached a strong degree of popularity in the 2000s.

Python is a high-level language, so it means that we don’t have to spend time thinking about several things like managing memory that we might need to think about in a lower-level language. This, also means that the language is more portable. Python can run on PC, Mac, or Linux because there is a separate software to install to provide access to it.

Python does have several advantages over many other languages. It is a relatively simple language that is easy to learn. Because Python is designed to be interpreted, it can provide the kind of runtime feedback that is especially helpful to novice programmers.

Python is dynamic or interpreted language because it will run our code line-by-line when we ask it to, without trying to compile it first. Alternative mode to this is a scripting mode, where we write a bunch of code then run it all at once.

### **Environment variables:**

An environment variable is a dynamic "object" on a computer that stores a value, which in turn can be referenced by one or more software programs in Windows. Environment variables help programs know what directory to install files in, where to store temporary files, where to find user profile settings, and other things. It can be said that environment variables help to create and shape the environment of where a program runs.

Two types of environment variables exist, user environment variables and system environment variables, as following:

**User environment variables**, as the name implies, are environment variables that are specific to each user account. This means that the value of an environment variable when logged in as one user can be different than the value of the same environment variable when logged in as a different user on the same computer.

**System environment variables** extend beyond just one user, applying to any user that might exist, or is created in the future. Most system environment variables point to important locations like the Windows folder.

### **Python Identifiers:**

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z OR a to z or an underscore (\_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, \$, and % within identifiers. Python is a case sensitive programming language. Thus, Manpower and manpower are two different identifiers in Python.

The following list shows the Python keywords. These are reserved words and you cannot use them as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only.

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

### **Comments in Python:**

A hash sign (#) that is not inside a string literal begins a comment. All characters after the # and up to the end of the physical line are part of the comment and the Python interpreter ignores them.

```
# This is a comment. x = "your name" # This is again  
comment # This is a comment, too.
```

### **Input from User:**

```
raw_input("Press the enter key to exit.") x =  
raw_input("Enter any number:")
```

### **Lab Activity:**

1. Download Python 2.7.14 from “www.python.org” website and install it by accepting default options.
2. To set environment variables, First, copy the address of python directory in C drive, by clicking on “This PC” and then go to the folder of “python27”. The address would be “C:\python27”.
3. After copying the address, right click on “This PC” and click on “Properties”. Click on “Advanced System Settings” and then click on “Advanced Tab”.
4. Click on “Environment Variables” and Under “System Variables”, search for “Path” and click on it. Then click on “Edit” and select “New”. Then paste the path of the python folder which is copied previously. Then click “ok”.
5. After setting the environment variable, create a folder “PythonPrograms” on the C drive, in which you store all your python programs.
6. Now type the “Notepad” in search field. In Notepad, write the programs as:

```
# This is my first python program in Hello.py file print 'Hello World!'
```

7. Now, go to “File” and click on “save”. In the save field, browse the C drive and select the “PythonPrograms” folder. In the “file name” field, remove everything and write the file name as “Hello.py”. In “Save as type” field, select “All Files”. Click on “save” to create your first python program.
8. To run the program, open the command prompt from search field, write “cd C:\” and press Enter.
9. Type “dir” to check the directories or files in C drive, here you should see the folder named as “PythonPrograms”, which you created previously.
10. Type “cd pythonprograms”, to change the directory from C drive to PythonPrograms folder.
11. Write “dir”, here you should see the file named as “Hello.py”.
12. To run the program, write “hello.py” and press Enter.
13. You should see the output Hello World!. Now you have created your first python program.

### **Lab Task 1:**

Write a program that displays your name, roll number and department name and attach print screen output of your program.

### **Lab Task 2:**

Write a program that create a pattern of stars as shown below using print command and attach print screen output of your program.

```
*  
**  
***  
****  
*****
```

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

**Lab Evaluation:**

<b>CLO3</b>	Apply logical problem-solving skills in writing computer programs for solving real life problems					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
						<b>Total</b>
						<b>/10</b>

**Lab No. 7 Objective:**

- To understand the Data Types and Variables in python language
- To learn how to perform Assignment, Arithmetic and comparison Operators □ To learn the use of relational and Boolean operators

**Pre-Lab Reading:**

Early computer programmers had the burdensome task of having to write their programs in the binary language of the machine they were programming. This means that computer instructions had to be hand-coded into binary numbers by the programmer before they could be entered into the machine. Furthermore, the programmers had to explicitly assign and reference any storage locations inside the computer's memory by a specific number or memory address. Today's programming languages allow you to concentrate more on solving the particular problem at hand than worrying about specific machine codes or memory locations. They enable you to assign symbolic names, known as variable names, for storing program computations and results.

A **Variable** name must be chosen by you in a meaningful way to reflect the type of value that is to be stored in that variable. The **assignment statement** creates new variables and assigns them values:

```
p = 10
a = "Introduction to Computing Lab"
```

The **assignment operator**, `=`, should not be confused with an equals sign. Assignment operators link a name, on the left-hand side of the operator, with a value, on the right-hand side.

**Data Types:**

Data types in Python programming language enables the programmers to appropriately select the data as per requirements of the program and the associated operations of handling it.

Python has five standard data types:

- Numbers
- String
- List
- Tuple
- Dictionary

Boolean (True/False) is also considered as data type in programming languages.

### **Numbers:**

Number data types store numeric values. Number objects are created when you assign a value to them.

Python supports four different numerical types –

- int (signed integers)
- long (long integers, they can also be represented in octal and hexadecimal)
- float (floating point real values) □ complex (complex numbers)

```
counter = 100          # An integer assignment miles    = 1000.0  
# A floating point
```

### **Strings:**

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

```
str = 'Hello World!' print str          # Prints complete string print  
str[0]      # Prints first character of the string print str[2:5]      #  
Prints characters starting from 3rd to 5th print str[2:]      # Prints  
string starting from 3rd character print str * 2      # Prints string two  
times print str + "TEST" # Prints concatenated string
```

### **Lists:**

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ] tinylist = [123,  
'john' ]
```

```
print list          # Prints complete list print list[0]      #
Prints first element of the list

print list[1:3]    # Prints elements starting from 2nd till 3rd  print
list[2:]         # Prints elements starting from 3rd element print tinylist * 2
# Prints list two times print list + tinylist # Prints concatenated lists
```

### Boolean:

Boolean data types are used to represent Boolean values True or False. This is used mostly to perform logical statements.

```
print (5==6)      # Prints boolean type
```

**Tuples and Dictionaries** will be discussed in coming labs.

### Value and Types:

A **value** is one of the fundamental things like a letter or a number that a program manipulates. The values we have seen as 2 and “Hello, World!”. These values belong to different **types**: 2 is an **integer**, and “Hello, World!” is a **string**. You can identify strings because they are enclosed in quotation marks. If you are not sure what type a value has, the **print(type())** statement can tell you as following:

```
print (type(2))           Output:<type 'int'> print (type (5.3))
Output:<type 'float'>

print (type (True))        Output:<type 'bool'> print (type ("Hello
world!"))
Output:<type 'str'>
```

### Operators:

Python language supports the following types of operators.

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

### Arithmetic Operators:

Assume variable a holds 10 and variable b holds 20, then

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	$a + b = 30$
- Subtraction	Subtracts right hand operand from left hand operand.	$a - b = -10$
* Multiplication	Multiplies values on either side of the operator	$a * b = 200$
/ Division	Divides left hand operand by right hand operand	$b / a = 2$
% Modulus	Divides left hand operand by right hand operand and returns remainder	$b \% a = 0$
** Exponent	Performs exponential (power) calculation on operators	$a^{**}b = 10 \text{ to the power } 20$
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) –	$9//2 = 4$ and $9.0//2.0 = 4.0$ , $-11//3 = -4$ , $-11.0//3 = -4.0$

### Comparison Operators:

These operators compare the values on either sides of them and decide the relation among them. They are also called Relational operators.

Assume variable a holds 10 and variable b holds 20, then

Operator	Description	Example
$==$	If the values of two operands are equal, then the condition becomes true.	$(a == b)$ is not true.
$!=$	If values of two operands are not equal, then condition becomes true.	$(a != b)$ is true.
$<>$	If values of two operands are not equal, then condition becomes true.	$(a <> b)$ is true. This is similar to $!=$ operator.

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.

### Assignment Operators:

Assume variable a holds 10 and variable b holds 20, then

Operator	Description	Example
=	Assigns values from right side operands to left side operand	c = a + b assigns value of a + b into c
+= Add AND	It adds right operand to the left operand and assign the result to left operand	c += a is equivalent to c = c + a
-= Subtract AND	It subtracts right operand from the left operand and assign the result to left operand	c -= a is equivalent to c = c - a
*= Multiply AND	It multiplies right operand with the left operand and assign the result to left operand	c *= a is equivalent to c = c * a
/= Divide AND	It divides left operand with the right operand and assign the result to left operand	c /= a is equivalent to c = c / a ac /= a is equivalent to c = c / a
%= Modulus AND	It takes modulus using two operands and assign the result to left operand	c %= a is equivalent to c = c % a

## University Of Engineering and Technology, Lahore Computer Engineering Department

**= Exponent AND	Performs exponential (power) calculation on operators and assign value to the left operand	c **= a is equivalent to c = c ** a
//= Floor Division	It performs floor division on operators and assign value to the left operand	c //= a is equivalent to c = c // a

### Logic Operators:

There are following logical operators supported by Python language. Assume variable a holds 10 and variable b holds 20 then

Operator	Description	Example
and Logical AND	If both the operands are true then condition becomes true.	(a and b) is true.
or Logical OR	If any of the two operands are non-zero then condition becomes true.	(a or b) is true.
not Logical NOT	Used to reverse the logical state of its operand.	Not (a and b) is false.

### Membership Operators:

Python's membership operators test for membership in a sequence, such as strings, lists, or tuples. There are two membership operators as explained below:

Operator	Description	Example
In	Evaluates to true if it finds a variable in the specified sequence and false otherwise.	x in y, here in results in a 1 if x is a member of sequence y.
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.	x not in y, here not in results in a 1 if x is not a member of sequence y.

### Identity Operators:

Identity operators compare the memory locations of two objects. There are two Identity operators explained below:

Operator	Description	Example
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.	x is y, here <b>is</b> results in 1 if id(x) equals id(y).
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise.	x is not y, here <b>is not</b> results in 1 if id(x) is not equal to id(y).

**Lab Tasks:**

1. Write a program that inputs miles from the user and converts miles into kilometers. One mile is equal to 1.609 kilometer.
2. Write a program that gets input two numbers from user, swaps the values and then displays them.  
 (Hint: use a temporary variable to help you swap values.)
3. Write a program that gets temperature from the user in Celsius and converts it into Fahrenheit using the formula  

$$F = \frac{9}{5}C + 32$$
4. Write a program that will prompt the user to enter the number of hours. It computes and displays the number of weeks, days and hours within the input number of hours.

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO2</b>	Understand general problem-solving strategies in development of computer algorithms.					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
<b>Total</b>						<b>/10</b>

### **Lab No. 8 Objective:**

- To understand the Decision Making in Python
- To learn how to perform If-else checks in Python Language

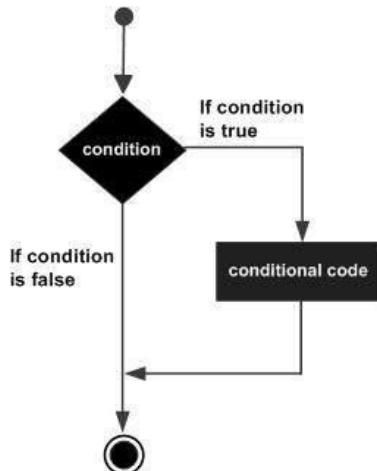
#### **If statement:**

The **if** statement is used to implement a decision. When a condition is fulfilled, one set of statements is executed. Otherwise, another set of statements is executed. **If** statement allows a program to carry out different actions depending on the nature of the data to be processed. In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly. **Conditional statements** give us this ability and the simplest form is the **if statement**:

```
x = 10 if x>0:    print 'x is
positive'
```

The Boolean expression after the **if statement** is called the **condition**. If it is true, then the indented statement gets executed. If not, nothing happens. The syntax for an **if statement** looks like this:

```
if expression:    statement(s)
```

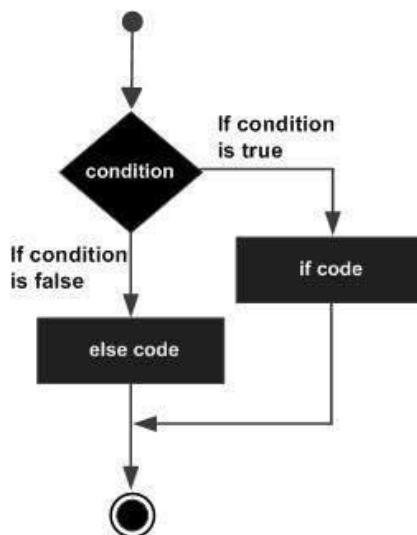


The **if statement** consists of a header and a body. The header begins with the keyword **if** followed by a Boolean expression and ends with a colon (:). The indented statements that follow are called a **block**. The first un-indented statement marks the end of the block. A statement block inside a compound statement is called the **body** of the statement. Each of the statements inside the body are executed in order if the Boolean expression evaluates to True. The entire block is skipped if the Boolean expression evaluates to False. There is no limit on the number of statements that can appear in the body of an if statement but there has to be at least one.

#### **If-else statement:**

A second form of the if statement is alternative execution, in which there are two possibilities and the condition determines which one gets executed. The syntax looks like this:

```
if expression:    statement(s)
else:      statement(s)
```



Consider the following example:

```
x = 10
if x %
2:
    print x, 'is even'
else:
    print x, 'is odd'
```

If the remainder when x is divided by 2 is 0, then we know that x is even, and the program displays a message to that effect. If the condition is false, the second set of statements is executed. Since the condition must be true or false, exactly one of the alternatives will be executed. The alternatives are called **branches**, because they are branches in the flow of execution.

#### **If-elif statement:**

Sometimes there are more than two possibilities and we need more than two branches. One way to express a computation like that is a **chained conditional**:

```
if expression1:    statement(s)
elif expression2:
statement(s) elif expression3:
statement(s) else:
statement(s)
```

elif is an abbreviation of else if. Again, exactly one branch will be executed. There is no limit to the number of elif statements but only a single (and optional) else statement is allowed and it must be the last branch in the statement. Each condition is checked in order. If the first is false, the next is checked and so on. If one of them is true, the corresponding branch executes, and the statement ends. Even if more than one condition is true, only the first true branch executes.

#### **Nested statement:**

One conditional can also be **nested** within another. Syntax of Nested statements looks like:

```
if expression1:    statement(s)
if expression2:
statement(s)    elif expression3:
statement(s)    elif expression4:
statement(s)    else:
                statement(s) else:
statement(s)
```

Consider the following example:

```
x = 20 y = 10 if x == y:    print x, ' and ', y, '
are equal' else:    if x > y:        print x, ' is
greater than ', y    else:        print x, ' is less
than ', y
```

The outer conditional contains two branches. The first branch contains a simple output statement. The second branch contains another if statement, which has two branches of its own. Those two branches are both output statements, although they could have been conditional statements as well. Although the indentation of the statements makes the structure apparent, nested conditionals become difficult to read very quickly. A nested conditional can always be rewritten as a chained conditional and vice versa. What to use is partly a matter of choice but in general you should use the structure that is easiest to understand.

Logical operators often provide a way to simplify nested conditional statements. For example, we can rewrite the following code using a single conditional:

```
x = 5 if 0 < x:    if x < 10:        print ' x is a
positive single digit '
```

The print statement is executed only if we make it past both the conditionals, so we can use the **and** operator:

```
x = 5 if 0 < x and x < 10:    print ' x is a positive  
single digit '
```

These kinds of conditions are common, so Python provides an alternative syntax that is similar to mathematical notation:

```
x = 5 if 0 < x < 10:    print ' x is a positive single  
digit '
```

This condition is semantically the same as the compound Boolean expression and the nested conditional.

### **String Formatting Operator:**

The **%** operator is used to format a set of variables enclosed in a "tuple" (a fixed size list), together with a format string, which contains normal text together with "argument specifiers", special symbols like "%s" and "%d". For example,

```
name = 'ali'  
age = 22  
print '%s is %d years old' %(name, age)
```

Here %s stands for string, %d for decimal numbers. We can also use %f for floating point number and %r for raw data.

### **Lab Tasks:**

1. Write a program that inputs two numbers and finds if second number is cube of first.
2. Write a program that inputs marks of three subjects. If the average of marks is more than 75, it displays two messages “you are above standard” and “Admission granted”.
3. Write a program that inputs salary and scale of your job. It adds 40% bonus if the scale is greater than 16. It adds 20% bonus if the scale is 16 or less and displays the total salary.
4. Write a program that inputs test score of a student and displays his grade according to the following criteria:

Test Score	Grade
>= 80	A
70 – 79	B

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

<b>60 – 69</b>	C
<b>50 – 59</b>	D
<b>Below 50</b>	F

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO2</b>	Understand general problem-solving strategies in development of computer algorithms.					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
						<b>Total</b>
						<b>/10</b>

### **Lab No. 9 Objective:**

- To understand the concept of loops in Python.
- To learn how to iterate the while-loop in Python.

### **Loops:**

In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. There may be a situation when you need to execute a block of code several number of times.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times.

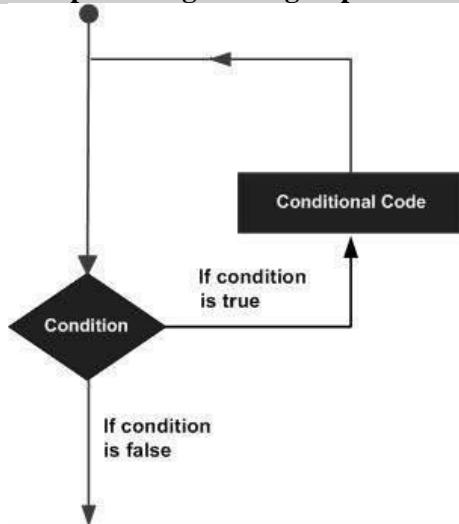
#### **While Loop:**

**While** looping statement repeats the number of lines until the condition is satisfied. A **while** loop is a control flow statement that allows code to be executed repeatedly based on a given **Boolean condition**. The **while** loop can be thought of as a repeating **if** statement.

#### **Syntax:**

```
while expression:    statement(s)
```

#### **Flow diagram:**



### Examples:

```
count = 0
while (count < 9):
    print 'The count is:', count
    count = count + 1
print "Good bye!"
```

The block here, consisting of the print and increment statements, is executed repeatedly until count is no longer less than 9. With each iteration, the current value of the index count is displayed and then increased by 1. Now consider the following example:

```
i = 0
x = None
while x != 0:
    x = input ("Enter any integer or press 0 for exit:")
    i = i + 1
print "Loop completed %d iterations before exit" %i
```

Write the above code and comment in last line which describes the output.

Following code helps you find the random number generated by random function of python:

```
import random
hiddenNumber = random.randint(1,100)
userGuess=0
while not userGuess == hiddenNumber:
    userGuess=int(raw_input("Guess a number: "))
    if userGuess > hiddenNumber and userGuess <= hiddenNumber+5:
        print"Close to hidden number but still high!"
    elif userGuess > hiddenNumber:
        print"Too high!"
    elif userGuess < hiddenNumber and userGuess >= hiddenNumber-5:
        print"Close to hidden number but still low!"
    elif userGuess < hiddenNumber:
        print"Too low!"
    else:
        print"That's right!"
```

Run this code in python and comment how this code helps you guess random number quickly.

### **Infinite Loop:**

A loop becomes infinite loop if a condition never becomes FALSE. You must use caution when using while loops because of the possibility that this condition never resolves to a FALSE value.

This results in a loop that never ends. Such a loop is called an infinite loop.

```
var = 0 while var == 0:      var = raw_input ("Enter  
a number")      print "You entered: ", num print  
"Good Bye"
```

Above example goes in an infinite loop and you need to use CTRL+C to exit the program.

### **Lab Tasks:**

1. Write a program that displays “Pakistan” for five times using **while** loop.
2. Write a program that inputs the count of numbers to calculate average. Then program asks user to input numbers one by one and displays average of all the numbers.

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO2</b>	Understand general problem-solving strategies in development of computer algorithms.					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
						<b>Total</b>
						<b>/10</b>

### **Lab No. 10 Objective:**

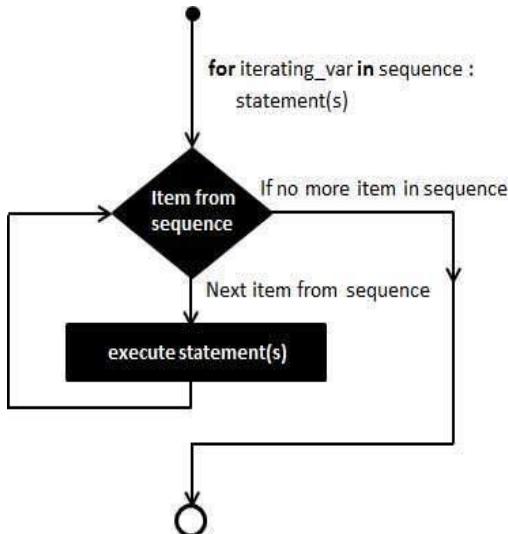
- To learn how to iterate the for-loop in Python.
- To understand for-loop with known and unknown ranges □ To learn how to use continue, break and pass statement **For Loop:**

Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. A **for** loop statement allows a number of lines represent until the condition is satisfied.

#### **Syntax:**

```
for iterating_var in sequence:  statements(s)
```

#### **Flow diagram:**



#### **Examples:**

Comment proper outputs of these examples:

```
for a in range(0,10): print a
```

Here, **range ()** function take three integers arguments as following: start, stop and step e.g. **range(start, stop, step)**. We can also use **range** function with only one argument as **range(stop)**. sum = 0.0 print “This program calculates the average of 5 numbers” for a in **range(0,10): print a**

Following example gives average of numbers input by users

```
sum = 0.0
count = int(raw_input("How many numbers you will average?"))
for i in range(1,count+1):
    num = float(raw_input("Enter numbers"+str(i)+":"))
    sum += num
print "Average:%0.4f"%(sum/count)
```

Another example of getting average of members of a list is give as:

```
list1=[10,20,30,40,50,60,70,80]
sum=0.0
for i in range(0,len(list1)):
    num=list1[i]
    sum+=num
print"Average: %0.2f"%(sum/len(list1))
```

## Nested For Loop:

The loops which contain another looping statement in a single loop. Any loop can contain a number of loop statements in itself.

### Syntax:

```
for iterating_var in sequence:
    for iterating_var in sequence:           statement(s)
```

### Examples:

```
Table=[2]
for i in Table:
    for j in range(1,11):
        print i,"*",j,"=",i*j
    print"\n"
```

Program for counting no. of spaces in members of list:

```
list2=["This is the first string","This is the Second string"]
spaces=0
for string1 in list2:
    for character1 in string1:
        if character1==" ":
            spaces+=1
print spaces
```

### Continue, break, pass and sleep statement:

**Continue** statement simply skips the remaining statements in the body of the loop, and continues with the next iteration of the loop.

In the following example print statement will be skipped when the number will be even.

```
for i in range(1, 21):
    if i% 2 == 0:
        continue
    print i, "is odd."
```

The **break** statement ends the loop and moves control to the next statement outside the loop.

```
for j in range(1, 10):
    for i in range(1, 10):
        if i% 2 == 0:
            break
        print i, "is odd."
```

The keyword **pass** is simply what you put if you want to run a loop that does nothing.

```
sequence= "ITCL"
for val in sequence:
    pass
```

The **sleep ()** function can be used to create delay.

```
import time
for i in range(1, 11):
    for j in range(1, 11):
        print i, "***",j, "=" , i* j
    time.sleep(2)
    print"\n"
```

### Lab Tasks:

1. Write a program that inputs a number from the user and displays a table of that number using **for loop**.

2. Write a program that inputs a number from the user and displays the factorial of that number using for loop.
3. Write a program that inputs a number from the user using **for loop**. It displays the number if the number is greater than 0 otherwise it inputs next number using continue statement.
4. Write a program that inputs a number from the user using **for loop**. It displays the number if the number is even and next number is input. Program exits the loop if the number is odd using break statement.
5. Write a program that displays the following block of stars using nested for loop.

```
*****
*****
*****
*****
*****
```

Hint: use >>> **print “\*”, <<<** instead of just using >>> **print “\*” <<<** to stop program from going in next line after every print command.

**Extra Task:**

Write a program that displays the following shape using nested for loops.

```
*
**
***
****
*****
```

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

<b>Course Name:</b>	<b>Course Code:</b>
<b>Assignment Type:</b> Assignment/Lab/Project	<b>Dated:</b>
<b>Semester:</b>	<b>Session:</b>
<b>Lab/Project/Assignment #:</b>	<b>CLOs to be covered:</b>
<b>Lab Title:</b>	<b>Teacher Name:</b>

### **Lab Evaluation:**

<b>CLO2</b>	Understand general problem-solving strategies in development of computer algorithms.					
<b>Levels (Marks)</b>	<b>Level1</b>	<b>Level2</b>	<b>Level3</b>	<b>Level4</b>	<b>Level5</b>	<b>Level6</b>
Psychomotor (8)						
Report Marks (2)						
						<b>Total</b>
						<b>/10</b>

### **Lab No. 11 Objective:**

- To understand the function concept in Python □ To learn how to define a function and how to call it

### **Pre-Lab Reading:**

#### **Function:**

It is a block of code that performs a specific task. It always return single value to the calling function. Functions are used to minimize the repetition of code. It can be executed from as many different parts as required. Optionally it returns a value to the calling program. There are some properties of a function as discussed below.

- Every function has a unique name. This name is used to call function. A function can be called from within another function.
- A function is independent and it can perform its task without intervention from or interfering with other parts of the program.
- A function performs a specific task. A task is a distinct job that a program must perform as a part of its overall operation, such as adding two or more integers, calculating a cube root, etc.
- A function returns a value to the calling program. This is optional and depends upon the task the function is going to accomplish. The syntax of a function is given below:

**def Function\_Name (Argument1, Argument2,.....):**

*Body of this Function*

*An optional return Statement(s)*

This program adds two numbers using user defined function **add** and return the output to the calling function.

```

def add(a,b):
    z=a+b
    return z
i,j=input("Enter two integer values separated by comma")
k=add(i,j)
print "addition of %d and %d using function=%d" % (i,j,k)

```

Now what happens, when you remove **return z** from the previous program, record and explain the output.

### **Function with no argument and no return value:**

The function without any arguments means that it cannot pass data to the called function. It is one of the simplest types of function. This type of function which does not return any value cannot be used in an expression and you can notice there will be no equality sign in it. This type of function can be used only as independent statement.

```
def printline():
    for i in range(0,30):
        print"-",
    print"\n"
print"Welcome to Functions in Python"
printline()
print"Functions are easy to learn"
printline()
```

### **Scoping:**

Each function defines a **new name** space, also called a **scope**. To keep track of which variables can be used, it is sometimes useful to store it in a frame, called **stack frame**. Each function is represented by a **frame**. A **frame** is a box with the name of a function beside it and the parameters and variables of the function inside it.

### **Lab activity:**

This program explains the use of **same variable** x, which has two different values according to its scope within the program and within the function.

```
def print_param(x):
    print x

x="hello"
print_param("Goodbye")
print x
```

### **Function call within another function:**

This program pass two strings as an argument to a function which is called in the main body of program. This function joins the two strings using the concatenation operator (+). The joined string is then passed as an argument to another function which is called inside the body of first function. The second function then prints the already joined strings twice.

*Note there is no return value in this program.*

```
def print_twice(phrase):
    print phrase,phrase

def print_joined_twice(part1,part2):
    joined = part1 + part2
    print_twice(joined)

line1= "Functions in"
line2= "Python"
print_joined_twice(line1,line2)
```

### **Lab Tasks:**

1. Write a function named **count\_down(a)** that takes an input integer value, **a**, as a parameter and count down until the value of '**a**' is greater than 0. When it gets to zero, it displays the phrase: **countdown is stopped**.

2. Write a function named **is\_even(n)** that takes an integer as an argument through input and returns **True** if the argument is an **even** and False if the argument is **odd**. If the returned value is True, your program will print **you entered even number**, and if the returned value is False your program will print **you entered an odd number**.
3. Write a function named **max(x, y)** that take two integer values as an argument and **return** the maximum value to a second function called **cube (z)**. This **cube(z)** function will **return** the cubic value of the maximum number, to the main body of program and also print this cubic value.