Beginners in Computer Science Studies

ISBN 978-93-90357-19-2

Website: www.skyfox.co

Email:skyfox@skyfox.org.in

Beginners in Computer Science Studies

sky fox

Beginners in COMPUTER SCIENCE STUDIES

Dr. Ignatious Herman. A
Mr. S. Nagarajan
Mrs. S. P. Sudha

Beginners in Computer Science Studies

AUTHOR INFORMATION:

Dr. Ignatious Herman A., B.Sc., M.Sc., Ph.D.,
Professor
Department of Computer Science & Engineering
DMI – St. Eugene University
P.O.Box: 330081, Chibombo
Zambia.

Mr. S. Nagarajan., M.E.,
Lecturer
Department of Computer Science & Engineering
DMI – St. Eugene University
P.O.Box: 330081, Chibombo
Zambia.

Mrs. S. P. Sudha., M.E.,
Lecturer
Department of Computer Science & Engineering
DMI – St. Eugene University
P.O.Box: 330081, Chibombo
Zambia.

First Edition	**: August 2020**
Cover Art and Design	**: Authors**
ISBN	**: 978-93-90357-19-2**
DOI	**: https://doi.org/10.22573/spg.020.BK/S/018**

Publisher Contact &Support
Skyfox Publishing Group
#987, Medical College Road
Thanjavur-613004
Tamil Nadu, India.
Phone: +918300123232
Email: skyfoxpublishing@gmail.com / skyfox@skyfox.org.in
Website: www.skyfox.co

Headquarters &Marketing Office

Skyfox Publishing Group
333 Cedar Street, PO Box 208002,
New Haven, United States.
CT 06520-8002.
Tel: 203.737.5603 / Fax: 203.785.7446
Email: skyfoxpublishing@gmail.com / skyfox@skyfox.org.in
Website: www.skyfox.co

Beginners in Computer Science Studies

Dr. Ignatious Herman. A

Mr. S. Nagarajan

Mrs. S. P. Sudha

From the desk of
Dr. T. X. A. ANANTH, BBA, MSW, MBA, MPhil, PhD,
President – University Council
Dear Learner,

Welcome to DMI – St. Eugene University!

I am sure you are expert in using the PC Tablets distributed by us. Now your world is open to Internet and using the tablet for your educational learning purposes. The very same book you are holding in your hand now is available in your V-Campus portal. All the teaching and learning materials are available in your portal.

As our Chancellor, Rev.Fr. Dr. J. E. Arulraj , mentioned, it is not just the success for DMI-St. Eugene University alone, it is success for the technology, it is success for the great nation of Zambia and it is success for the continent of Africa.

You can feel the improvement in the quality of the learning materials in the printed format. Improvement is done in quality of the content and the printing. Use it extensively and preserve it for your future references. This will help you to understand the subjects in a better way. The theories of the subjects have been explained thoroughly and the problems have been identified at learners' level. This book is made up of five units and every unit is mapped to the syllabus and discussed in detail.

I am happy at the efforts taken by the University in publishing this book not only in printed format, but also in PDF format in the Internet.

With warm regards

Dr. T. X. A. ANANTH
President – University Council

Chapter -1
COMPUTER STUDIES –II

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language. It keeps fluctuating at number one scale of popularity along with Java programming language, which is also equally popular and most widely used among modern software programmers.

C programming language is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Software Development Domain. I will list down some of the key advantages of learning C Programming:
Easy to learn

- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

Facts about C

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around the early 1970s.
- The language was formalized in 1988 by the American National Standard Institute (ANSI).
- The UNIX OS was totally written in C.
- Today C is the most widely used and popular System Programming Language.
- Most of the state-of-the-art software have been implemented using C.

- Today's most popular Linux OS and RDBMS MySQL have been written in C.

C Programming.

Just to give you a little excitement about **C programming,** sample program

```
#include <stdio.h>

void main() {
  /* my first program in C */
  printf("Welcome\n");

}
```

C - Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators –

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Misc Operators

We will, in this chapter, look into the way each operator works.

Arithmetic Operators

The following table shows all the arithmetic operators supported by the C language. Assume variable **A** holds 15 and variable **B** holds 30 then

Operator	Description	Example
+	Adds two operands.	A + B = 45
–	Subtracts second operand from the first.	A – B = -15
*	Multiplies both operands.	A * B = 450
/	Divides numerator by de-numerator.	B / A = 2
%	Modulus Operator and remainder of after an integer division.	B % A = 0
++	Increment operator increases the integer value by one.	A++ = 16
--	Decrement operator decreases the integer value by one.	A-- = 14

Relational Operators

The following table shows all the relational operators supported by C. Assume variable **A** holds 10 and variable **B** holds 20 then

Operator	Description	Example
==	Checks if the values of two operands are equal or not. If yes, then the condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not. If the values are not equal, then the condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand. If yes, then the condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand. If yes, then the condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand. If yes, then the condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to	(A <= B) is

	the value of right operand. If yes, then the condition becomes true.	true.

Logical Operators

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	!(A && B) is true.

Following table shows all the logical operators supported by C language. Assume variable **A** holds 1 and variable **B** holds 0, then –

Show Examples

Bitwise Operators

Bitwise operator works on bits and perform bit-by-bit operation. The truth tables for &, |, and ^ is as follows –

p	q	p & q	p | q	p ^ q

0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

Assume A = 60 and B = 13 in binary format, they will be as follows –

A = 0011 1100

B = 0000 1101

A&B = 0000 1100

A|B = 0011 1101

A^B = 0011 0001

~A = 1100 0011

The following table lists the bitwise operators supported by C. Assume variable 'A' holds 60 and variable 'B' holds 13, then –

Example

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100

|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. -0111101
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Assignment Operators

The following table lists the assignment operators supported by the C language –

Examples

C Input and Output

Input means to provide the program with some data to be used in the program and Output means to display data on screen or write the data to a printer or a file.
C programming language provides many built-in functions to read any given input and to display data on screen when there is a need to output the result.
All these built-in functions are present in C header files, we will also specify the name of header files in which a particular function is defined while discussing about it.

scanf() and **printf()** functions
The standard input-output header file, named stdio.h contains the definition of the functions printf() and scanf(), which are used to display output on screen and to take input from user respectively.
#include<stdio.h>

```
void main()
{
   // defining a variable
   int i;
   /*
      displaying message on the screen
      asking the user to input a value
   */
   printf("Please enter a value...");
   /*
      reading the value entered by the user
   */
   scanf("%d", &i);
   /*
      displaying the number as output
   */
   printf( "\nYou entered: %d", i);
}
```

When you will compile the above code, it will ask you to enter a value. When you will enter the value, it will display the value you have entered on screen.

You must be wondering what is the purpose of %d inside the scanf() or printf() functions. It is known as format string and this informs the scanf() function, what type of input to expect and in printf() it is used to give a heads up to the compiler, what type of output to expect.

Format String	Meaning
%d	Scan or print an integer as signed decimal number
%f	Scan or print a floating point number
%c	To scan or print a character
%s	To scan or print a character string. The scanning ends at whitespace.

We can also limit the number of digits or characters that can be input or output, by adding a number with the format string specifier, like "%1d" or "%3s", the first one means a single numeric digit and the second one means 3 characters, hence if you try to input 42, while scanf() has "%1d", it will take only 4 as input. Same is the case for output.

getchar() & putchar() functions

The getchar() function reads a character from the terminal and returns it as an integer. This function reads only single character at a time. You can use this method in a loop in case you want to read more than one character. The putchar() function displays the character passed to it on the screen and returns the same character. This function too displays only a single character at a time. In case you want to display more than one characters, use putchar() method in a loop.

```
#include <stdio.h>
void main( )
{
   int c;
   printf("Enter a character");

/*  Take a character as input and store it in variable c */
   c = getchar();

   /*display the character stored in variable c */
   putchar(c);
}
```

gets() & puts() functions

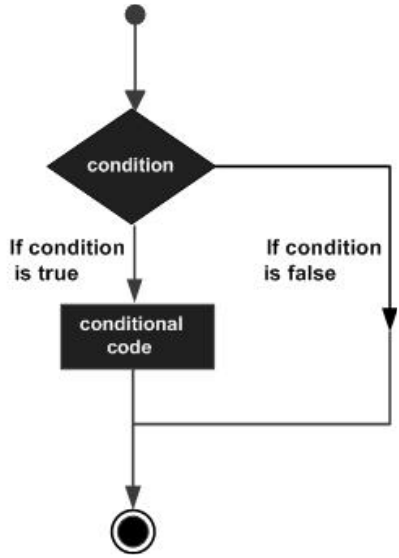
The gets() function reads a line from stdin(standard input) into the buffer pointed to by str pointer, until either a terminating newline or EOF (end of file) occurs. The puts() function writes the string str and a trailing newline to stdout.

```
#include<stdio.h>
void main()
{
   /* character array of length 100 */
   char str[100];
   printf("Enter a string");
   gets( str );
   puts( str );
   getch();
}
```

CONTROL STATEMENTS

Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Show below is the general form of a typical decision making structure found in most of the programming languages



C programming language assumes any non-zero and non-null values as true, and if it is either zero or null, then it is assumed as false value.

S.No.	Statement & Description
1	if statement An **if statement** consists of a boolean expression followed by

	one or more statements.
2	if...else statement An **if statement** can be followed by an optional **else statement**, which executes when the Boolean expression is false.
3	nested if statements You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).
4	switch statement A **switch** statement allows a variable to be tested for equality against a list of values.
5	nested switch statements You can use one **switch** statement inside another **switch** statement(s).

C programming language provides the following types of decision making statements.

An if statement consists of a Boolean expression followed by one or more statements.

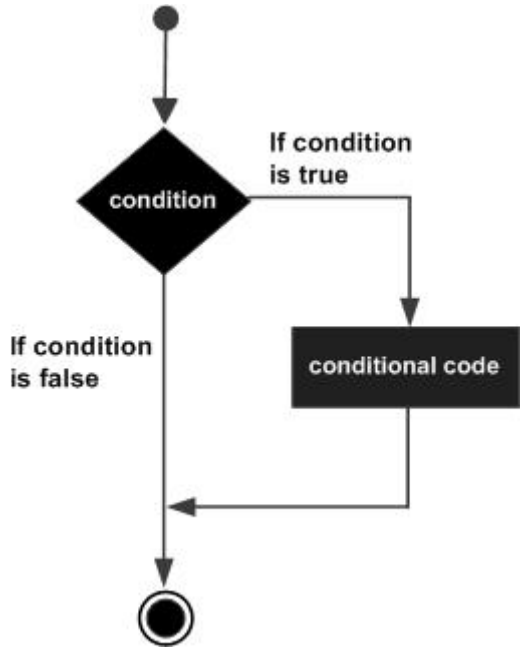
Syntax

The syntax of an 'if' statement in C programming language is –

```
if(boolean_expression)
{
  /* statement(s) will execute if the boolean expression is true */
}
```

If the Boolean expression evaluates to true, then the block of code inside the 'if' statement will be executed. If the Boolean expression evaluates to false, then the first set of code after the end of the 'if' statement (after the closing curly brace) will be executed.
C programming language assumes any non-zero and non-null values as true and if it is either zero or null, then it is assumed as false value.

Flow Diagram



An **if** statement can be followed by an optional **else** statement, which executes when the Boolean expression is false.

Syntax

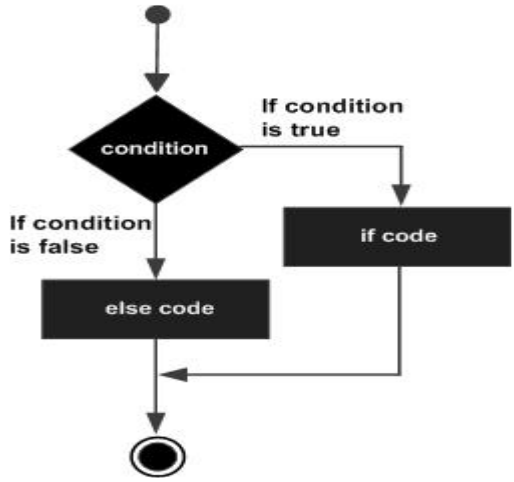
The syntax of an **if...else** statement in C programming language is –

```
if(boolean_expression) {
   /* statement(s) will execute if the boolean expression is true */
} else {
   /* statement(s) will execute if the boolean expression is false */
}
```

If the Boolean expression evaluates to **true**, then the **if block** will be executed, otherwise, the **else block** will be executed.

C programming language assumes any non-zero and non-null values as true, and if it is either zero or null, then it is assumed as false value

Flow Diagram



Example

```
#include <stdio.h>

int main () {

   /* local variable definition */
   int a = 100;
```

```
  /* check the boolean condition */
  if( a < 20 ) {
    /* if condition is true then print the following */
    printf("a is less than 20\n" );
  } else {
    /* if condition is false then print the following */
    printf("a is not less than 20\n" );
  }

  printf("value of a is : %d\n", a);

  return 0;
}
```

If...else if...else Statement

An **if** statement can be followed by an optional **else if...else** statement, which is very useful to test various conditions using single if...else if statement.

When using if...else if..else statements, there are few points to keep in mind –

- An if can have zero or one else's and it must come after any else if's.
- An if can have zero to many else if's and they must come before the else.
- Once an else if succeeds, none of the remaining else if's or else's will be tested.

Syntax

The syntax of an **if...else if...else** statement in C programming language is –

```
if(boolean_expression 1) {
  /* Executes when the boolean expression 1 is true */
```

```
} else if( boolean_expression 2) {
   /* Executes when the boolean expression 2 is true */
} else if( boolean_expression 3) {
   /* Executes when the boolean expression 3 is true */
} else {
   /* executes when the none of the above condition is true */
}
```

Example

```
#include <stdio.h>

int main () {

   /* local variable definition */
   int a = 100;

   /* check the boolean condition */
   if( a == 10 ) {
      /* if condition is true then print the following */
      printf("Value of a is 10\n" );
   } else if( a == 20 ) {
      /* if else if condition is true */
      printf("Value of a is 20\n" );
   } else if( a == 30 ) {
      /* if else if condition is true  */
      printf("Value of a is 30\n" );
   } else {
      /* if none of the conditions is true */
      printf("None of the values is matching\n" );
   }

   printf("Exact value of a is: %d\n", a );

   return 0;
```

```
}
```

When the above code is compiled and executed, it produces the following result –

```
None of the values is matching
Exact value of a is: 100
```

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case**.

Syntax

The syntax for a **switch** statement in C programming language is as follows –

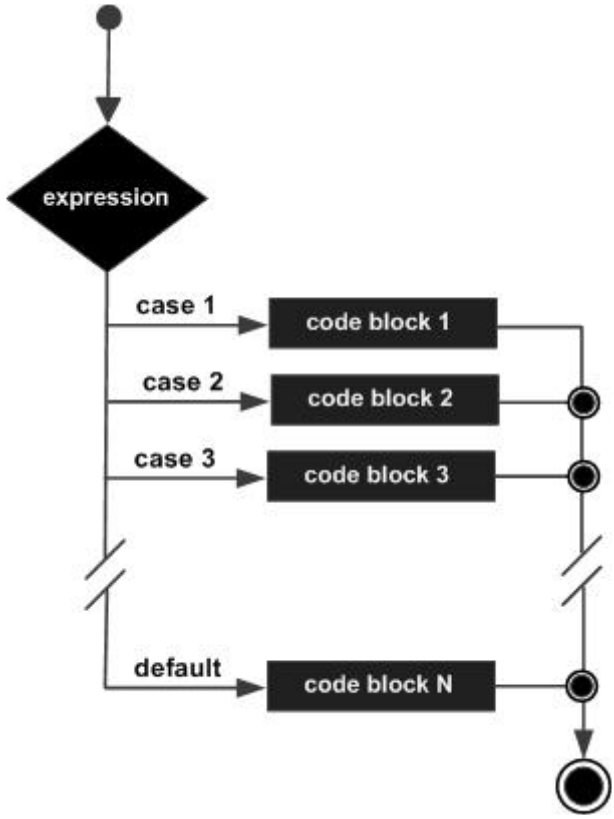
```
switch(expression){

case constant-expression  :
   statement(s);
break;/* optional */

case constant-expression  :
   statement(s);
break;/* optional */

/* you can have any number of case statements */
default:/* Optional */
  statement(s);
}
```

The following rules apply to a **switch** statement –

- The **expression** used in a **switch** statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.

- The **constant-expression** for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a **break** statement is reached.
- When a **break** statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a **break**. If no **break** appears, the flow of control will *fall through* to subsequent cases until a break is reached.
- A **switch** statement can have an optional **default** case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No **break** is needed in the default case.

  Flow Diagram



Example

```
#include <stdio.h>

int main () {

  /* local variable definition */
  char grade = 'B';

  switch(grade) {
    case 'A' :
      printf("Excellent!\n" );
      break;
```

```
    case 'B' :
    case 'C' :
      printf("Well done\n" );
      break;
    case 'D' :
      printf("You passed\n" );
      break;
    case 'F' :
      printf("Better try again\n" );
      break;
    default :
      printf("Invalid grade\n" );
  }

  printf("Your grade is  %c\n", grade );

  return 0;
}
```

When the above code is compiled and executed, it produces the following result –

```
Well done
Your grade is B
```

The ? : Operator

We have covered conditional operator ? : in the previous chapter which can be used to replace if...else statements. It has the following general form –

Exp1? Exp2 : Exp3;

Where Exp1, Exp2, and Exp3 are expressions. Notice the use and placement of the colon.

The value of a? expression is determined like this –

Exp1 is evaluated. If it is true, then Exp2 is evaluated and becomes the value of the entire ? expression.

If Exp1 is false, then Exp3 is evaluated and its value becomes the value of the expression.

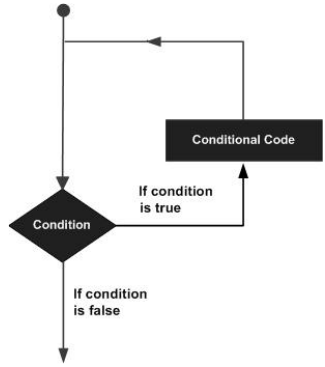
C - Loops

When a block of code needs to be executed several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.

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. Given below is the general form of a loop statement in most of the programming languages

C programming language provides the following types of loops to handle looping requirements.



Sr.No.	Loop Type & Description
1	while loop Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
2	for loop Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
3	do...while loop It is more like a while statement, except that it tests the condition at the end of the loop body.
4	nested loops You can use one or more loops inside any other while, for, or do..while loop.

A **while** loop in C programming repeatedly executes a target statement as long as a given condition is true.

Syntax

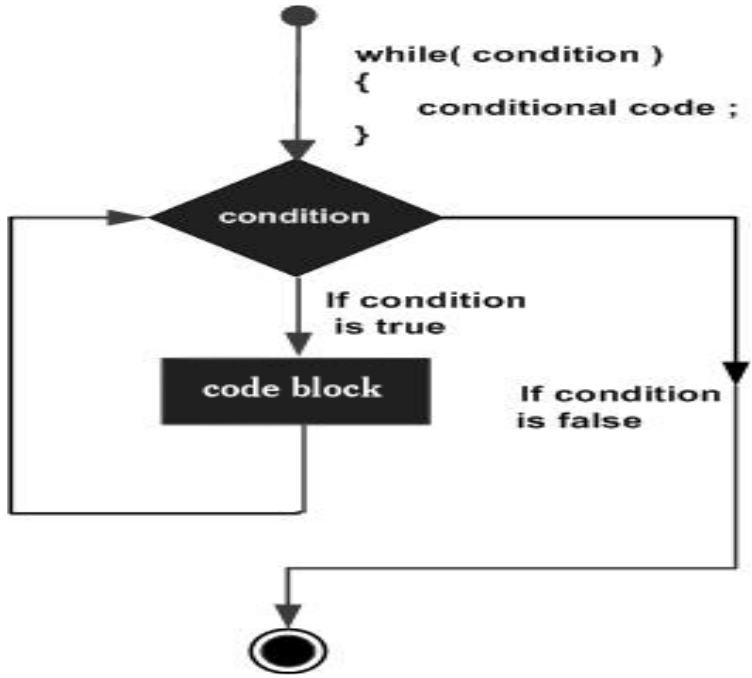
The syntax of a **while** loop in C programming language is –

```
while(condition)
{
  statement(s);
}
```

Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any nonzero value. The loop iterates while the condition is true.

When the condition becomes false, the program control passes to the line immediately following the loop.

Flow Diagram



Here, the key point to note is that a while loop might not execute at all. When the condition is tested and the result is false, the loop body will be skipped and the first statement after the while loop will be executed.

Example

```
#include <stdio.h>

int main () {

  /* local variable definition */
```

```
  int a = 10;

  /* while loop execution */
  while( a < 20 ) {
    printf("value of a: %d\n", a);
    a++;
  }

  return 0;
}
```

When the above code is compiled and executed, it produces the following result –

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

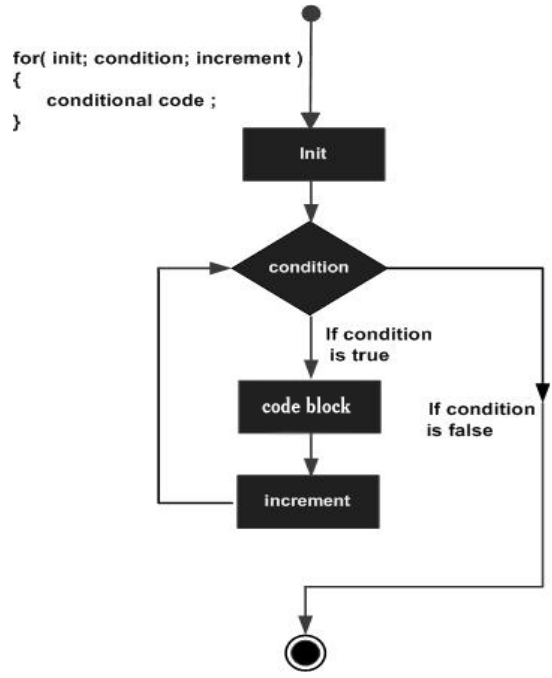
Syntax

The syntax of a **for** loop in C programming language is –

```
for ( init; condition; increment ) {
  statement(s);
}
```

Here is the flow of control in a 'for' loop –

- The **init** step is executed first, and only once. This step allows you to declare and initialize any loop control variables. You are not required to put a statement here, as long as a semicolon appears.
- Next, the **condition** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and the flow of control jumps to the next statement just after the 'for' loop.
- After the body of the 'for' loop executes, the flow of control jumps back up to the **increment** statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.
- The condition is now evaluated again. If it is true, the loop executes and the process repeats itself (body of loop, then increment step, and then again condition). After the condition becomes false, the 'for' loop terminates.



Flow Diagram

Example

```
#include<stdio.h>

int main (){

int a;

/* for loop execution */
for( a =10; a <20; a = a +1){
    printf("value of a: %d\n", a);
}

return0;
}
```

When the above code is compiled and executed, it produces the following result –

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

do...while loop in C

Unlike **for** and **while** loops, which test the loop condition at the top of the loop, the **do...while** loop in C programming checks its condition at the bottom of the loop.

A **do...while** loop is similar to a while loop, except the fact that it is guaranteed to execute at least one time.

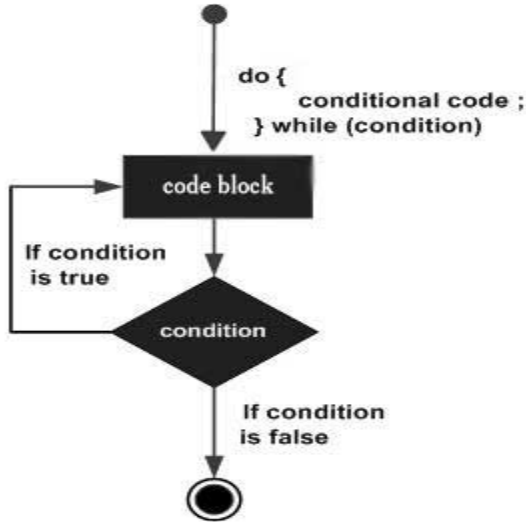
Syntax

The syntax of a **do...while** loop in C programming language is –

```
do {
  statement(s);
} while( condition );
```

Notice that the conditional expression appears at the end of the loop, so the statement(s) in the loop executes once before the condition is tested. If the condition is true, the flow of control jumps back up to do, and the statement(s) in the loop executes again. This process repeats until the given condition becomes false.

Flow Diagram



Example

```
#include<stdio.h>

int main (){

/* local variable definition */
```

```
int a =10;

/* do loop execution */
do{
    printf("value of a: %d\n", a);
    a = a +1;
}while( a <20);

return0;
}
```

When the above code is compiled and executed, it produces the following result –

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

FUNCTIONS

A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.

A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.

Defining a Function

The general form of a function definition in C programming language is as follows –

```
return_type function_name( parameter list ){
  body of the function
}
```

A function definition in C programming consists of a *function header* and a *function body*. Here are all the parts of a function –

- **Return Type** – A function may return a value. The **return_type** is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword **void**.
- **Function Name** – This is the actual name of the function. The function name and the parameter list together constitute the function signature.
- **Parameters** – A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.
- **Function Body** – The function body contains a collection of statements that define what the function does.

Example

Given below is the source code for a function called **max()**. This function takes two parameters num1 and num2 and returns the maximum value between the two –

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

/* local variable declaration */
```

```
int result;

if(num1 > num2)
    result = num1;
else
    result = num2;

return result;
}
```

Arrays in C

In C language, arrays are reffered to as structured data types. An array is defined as finite ordered collection of homogenous data, stored in contiguous memory locations.

Here the words,

- finite means data range must be defined.
- ordered means data must be stored in continuous memory addresses.
- homogenous means data must be of similar data type.

Example where arrays are used,

- to store list of Employee or Student names,
- to store marks of students,
- or to store list of numbers or characters etc.

Since arrays provide an easy way to represent data, it is classified amongst the data structures in C. Other data structures in c are structure, lists, queues, trees etc. Array can be used to represent not only simple list of data but also table of data in two or three dimensions.

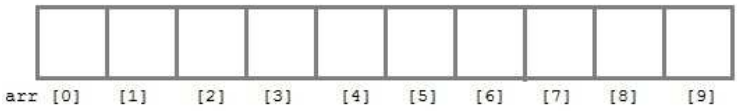
Declaring an Array

Like any other variable, arrays must be declared before they are used. General form of array declaration is,

data-type variable-name[size];

/* Example of array declaration */

int arr[10];
array declaraction in c



Here int is the data type, arr is the name of the array and 10 is the size of array. It means array arr can only contain 10 elements of int type.

Index of an array starts from 0 to size-1 i.e first element of arr array will be stored at arr[0] address and the last element will occupy arr[9].

Initialization of an Array

After an array is declared it must be initialized. Otherwise, it will contain **garbage** value(any random value). An array can be initialized at either **compile time** or at **runtime**.

Compile time Array initialization

Compile time initialization of array elements is same as ordinary variable initialization. The general form of initialization of array is,

```
data-type array-name[size] = { list of values };
/* Here are a few examples */
int marks[4]={ 57, 87, 56, 77 };    // integer array initialization
float area[5]={ 43.4, 6.8, 5.5 };   // float array initialization
int marks[4]={ 67, 47, 56, 77, 69 };    // Compile time error
```

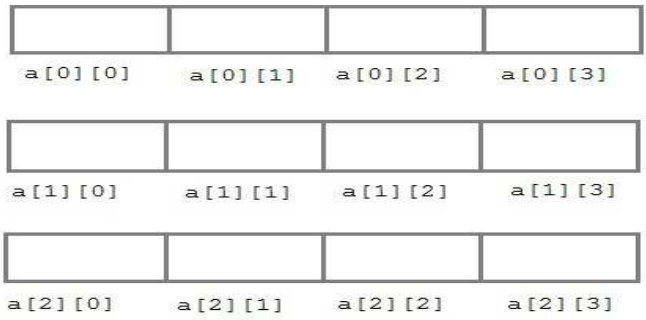
Two dimensional Arrays

C language supports multidimensional arrays also. The simplest form of a multidimensional array is the two-dimensional array. Both the row's and column's index begins from 0.

Two-dimensional arrays are declared as follows

data-type array-name[row-size][column-size]

```
/* Example */
int a[3][4];
```



An array can also be declared and initialized together. For example,

```
int arr[][3] = {
   {0,0,0},
   {1,1,1}
};
```

Runtime initialization of a two dimensional Array

```
#include<stdio.h>
void main()
{
   int arr[3][4];
   int i, j, k;
   printf("Enter array element");
   for(i = 0; i < 3;i++)
   {
      for(j = 0; j < 4; j++)
      {
         scanf("%d", &arr[i][j]);
      }
   }
   for(i = 0; i < 3; i++)
   {
      for(j = 0; j < 4; j++)
      {
         printf("%d", arr[i][j]);
      }
   }
}
```

Chapter -II

String and Character Array

String is a sequence of characters that is treated as a single data item and terminated by null character '\0'. Remember that C language does not support strings as a data type. A string is actually one-dimensional array of characters in C language. These are often used to create meaningful and readable programs.

For example: The string "hello world" contains 12 characters including '\0' character which is automatically added by the compiler at the end of the string.
Declaring and Initializing a string variables

There are different ways to initialize a character array variable.

```
char name[13] = "StudyTonight";      // valid character array initialization

char name[10] = {'L','e','s','s','o','n','s','\0'};    // valid initialization
```

Remember that when you initialize a character array by listing all of its characters separately then you must supply the '\0' character explicitly.

Some examples of illegal initialization of character array are,

```
char ch[3] = "hell";   // Illegal



char str[4];

str = "hell";  // Illegal
```

String Input and Output

Input function scanf() can be used with %s format specifier to read a string input from the terminal. But there is one problem with scanf() function, it terminates its input on the first white space it encounters.

Therefore if you try to read an input string "Hello World" using scanf() function, it will only read Hello and terminate after encountering white spaces.
However, C supports a format specification known as the edit set conversion code %[..] that can be used to read a line containing a variety of characters, including white spaces.

```
#include<stdio.h>

#include<string.h>

void main()

{

   char str[20];

   printf("Enter a string");

   scanf("%[^\n]", &str);  //scanning the whole string, including the white spaces

   printf("%s", str);

}
```

Another method to read character string with white spaces from terminal is by using the gets() function.

```
char text[20];

gets(text);

printf("%s", text);
```

String Handling Functions

C language supports a large number of string handling functions that can be used to carry out many of the string manipulations. These

functions are packaged in **string.h** library. Hence, you must include **string.h** header file in your programs to use these functions.
The following are the most commonly used string handling functions.

Method	Description
strcat()	It is used to concatenate(combine) two strings
strlen()	It is used to show length of a string
strrev()	It is used to show reverse of a string
strcpy()	Copies one string into another
strcmp()	It is used to compare two string

strcat() function

strcat("hello", "world");

strcat() function will add the string "world" to "hello" i.e it will ouput helloworld.

strlen() function

strlen() function will return the length of the string passed to it.

```
int j;

j = strlen("studytonight");

printf("%d",j);
```

Output

12

strcmp() function

strcmp() function will return the ASCII difference between first unmatching character of two strings.

```
int j;

j = strcmp("study", "tonight");

printf("%d",j);
```

Output

-1

strcpy() function

It copies the second string argument to the first string argument.

```
#include<stdio.h>
#include<string.h>

int main()
{
   char s1[50];
   char s2[50];
```

```
   strcpy(s1, "StudyTonight");    //copies "studytonight" to string s1
   strcpy(s2, s1);    //copies string s1 to string s2

   printf("%s\n", s2);

   return(0);
}
```

Output

StudyTonight

strrev() function

It is used to reverse the given string expression.

```
#include<stdio.h>

int main()
{
   char s1[50];

   printf("Enter your string: ");
   gets(s1);
   printf("\nYour reverse string is: %s",strrev(s1));
   return(0);
}
```

Output

Enter your string: studytonight

Your reverse string is: thginotyduts

C - Pointers

Pointers in C are easy and fun to learn. Some C programming tasks are performed more easily with pointers, and other tasks, such as dynamic memory allocation, cannot be performed without using pointers. So it

becomes necessary to learn pointers to become a perfect C programmer. Let's start learning them in simple and easy steps.

As you know, every variable is a memory location and every memory location has its address defined which can be accessed using ampersand (&) operator, which denotes an address in memory. Consider the following example, which prints the address of the variables defined

```
#include <stdio.h>

int main () {

  int  var1;
  char var2[10];

  printf("Address of var1 variable: %x\n", &var1  );
  printf("Address of var2 variable: %x\n", &var2  );

  return 0;
}
```

When the above code is compiled and executed, it produces the following result –

```
Address of var1 variable: bff5a400
Address of var2 variable: bff5a3f6
```

What are Pointers?

A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. The general form of a pointer variable declaration is

type *var-name;

Here, type is the pointer's base type; it must be a valid C data type and var-name is the name of the pointer variable. The asterisk * used to declare a pointer is the same asterisk used for multiplication. However,

in this statement the asterisk is being used to designate a variable as a pointer. Take a look at some of the valid pointer declarations

```
int    *ip;    /* pointer to an integer */

double *dp;    /* pointer to a double */

float  *fp;    /* pointer to a float */

char   *ch     /* pointer to a character */
```

The actual data type of the value of all pointers, whether integer, float, character, or otherwise, is the same, a long hexadecimal number that represents a memory address. The only difference between pointers of different data types is the data type of the variable or constant that the pointer points to.

How to Use Pointers?

There are a few important operations, which we will do with the help of pointers very frequently. (a) We define a pointer variable, (b) assign the address of a variable to a pointer and (c) finally access the value at the address available in the pointer variable. This is done by using unary operator * that returns the value of the variable located at the address specified by its operand. The following example makes use of these operations –

```
#include <stdio.h>

int main () {

  int  var = 20;   /* actual variable declaration */
  int  *ip;        /* pointer variable declaration */

  ip = &var;  /* store address of var in pointer variable*/

  printf("Address of var variable: %x\n", &var  );

  /* address stored in pointer variable */
```

```
   printf("Address stored in ip variable: %x\n", ip );

   /* access the value using the pointer */
   printf("Value of *ip variable: %d\n", *ip );

   return 0;
}
```

When the above code is compiled and executed, it produces the following result –

```
Address of var variable: bffd8b3c
Address stored in ip variable: bffd8b3c
Value of *ip variable: 20
```

NULL Pointers

It is always a good practice to assign a NULL value to a pointer variable in case you do not have an exact address to be assigned. This is done at the time of variable declaration. A pointer that is assigned NULL is called a null pointer.

The NULL pointer is a constant with a value of zero defined in several standard libraries. Consider the following program

```
#include <stdio.h>

int main () {

   int  *ptr = NULL;

   printf("The value of ptr is : %x\n", ptr  );

   return 0;
}
```

When the above code is compiled and executed, it produces the following result –

The value of ptr is 0

In most of the operating systems, programs are not permitted to access memory at address 0 because that memory is reserved by the operating system. However, the memory address 0 has special significance; it signals that the pointer is not intended to point to an accessible memory location. But by convention, if a pointer contains the null (zero) value, it is assumed to point to nothing.

To check for a null pointer, you can use an 'if' statement as follows –

```
if(ptr)     /* succeeds if p is not null */
if(!ptr)    /* succeeds if p is null */
```

Pointers in Detail

Pointers have many but easy concepts and they are very important to C programming. The following important pointer concepts should be clear to any C programmer –

Concept & Description
Pointer arithmetic There are four arithmetic operators that can be used in pointers: ++, --, +, -
Array of pointers You can define arrays to hold a number of pointers.
Pointer to pointer C allows you to have pointer on a pointer and so on.

Passing pointers to functions in C

Passing an argument by reference or by address enable the passed argument to be changed in the calling function by the called function.

Return pointer from functions in C

C allows a function to return a pointer to the local variable, static variable, and dynamically allocated memory as well.

Chapter III

Boolean Logic

Boolean Algebra is the mathematics we use to analyse digital gates and circuits. We can use these "Laws of Boolean" to both reduce and simplify a complex Boolean expression in an attempt to reduce the number of logic gates required. Boolean Algebra is therefore a system of mathematics based on logic that has its own set of rules or laws which are used to define and reduce Boolean expressions.

The variables used in Boolean Algebra only have one of two possible values, a logic "0" and a logic "1" but an expression can have an infinite number of variables all labelled individually to represent inputs to the expression, For example, variables A, B, C etc, giving us a logical expression of A + B = C, but each variable can ONLY be a 0 or a 1.

Examples of these individual laws of Boolean, rules and theorems for Boolean Algebra are given in the following table.
Boolean Algebra deals mainly with the theory that both logic and set operations are either "TRUE" or "FALSE" but not both at the same time.

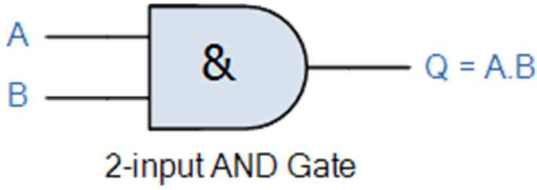
For example, A + A = A and not 2A as it would be in normal algebra. Boolean Algebra is a simple and effective way of representing the switching action of standard Logic Gates and the basic logic statements which concern us here are given by the logic gate operations of the AND, the OR and the NOT gate functions.

The **logic AND** Function
The Logic AND Function function states that two or more events must occur together and at the same time for an output action to occur. The order in which these actions occur is unimportant as it does not affect the final result. For example, A & B = B & A. In Boolean algebra the Logic AND Function follows the Commutative Law which allows a change in position of either variable.

The AND function is represented in electronics by the dot or full stop symbol (.) Thus a 2-input (A B) AND Gate has an output term represented by the Boolean expression A.B or just AB.

AND Function Truth Table

Switch A	Switch B	Output	Description
0	0	0	A and B are both open, lamp OFF
0	1	0	A is open and B is closed, lamp OFF
1	0	0	A is closed and B is open, lamp OFF
1	1	1	A is closed and B is closed, lamp ON
Boolean Expression (A AND B)			A . B



2-input AND Gate

The Logic OR Function function states that an output action will become TRUE if either one "OR" more events are TRUE, but the order at which they occur is unimportant as it does not affect the final result.

For example, A + B = B + A. In Boolean algebra the Logic OR Function follows the Commutative Law the same as for the logic AND function, allowing a change in position of either variable.

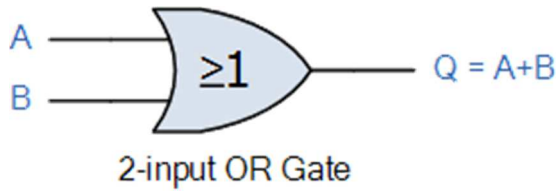
The OR function is sometimes called by its full name of "Inclusive OR" in contrast to the Exclusive-OR function we will look at later in tutorial six.

The logic or Boolean expression given for a logic OR gate is that for Logical Addition which is denoted by a plus sign, (+). Thus a 2-input (A B) Logic OR Gate has an output term represented by the Boolean expression of: A+B = Q.

OR Function Truth Table

Switch A	Switch B	Output	Description
0	0	0	A and B are both open, lamp OFF

0	1	1	A is open and B is closed, lamp ON
1	0	1	A is closed and B is open, lamp ON
1	1	1	A is closed and B is closed, lamp ON
Boolean Expression (A OR B)			A + B



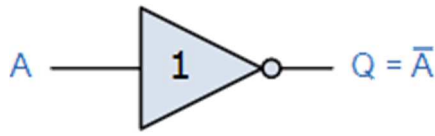
2-input OR Gate

The **"logic NOT function"** is so called because its output state is NOT the same as its input state with its Boolean Expression generally denoted by a bar or overline (¯) over its input symbol which denotes the inversion operation, (hence its name as an inverter).

If A means that the switch is closed, then NOT A or simply A says that the switch is **NOT** closed or in other words, it is open. The logic NOT function has a single input and a single output as shown.

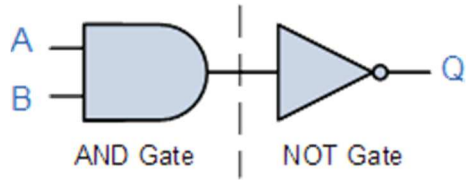
NOT Function Truth Table

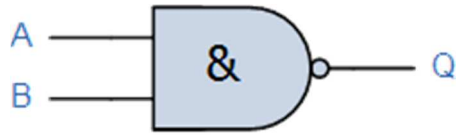
Switch	Output
1	0
0	1
Boolean Expression	not-A or A



Inverter or NOT Gate

The **NAND or "Not AND**" function is a combination of the two separate logical functions, the AND function and the NOT function in series. The logic NAND function can be expressed by the Boolean expression of, A.B 2-input logic gate





2-input NAND Gate

The truth table for the NAND function is the opposite of that for the previous AND function because the NAND gate performs the reverse operation of the AND gate. In other words, the NAND gate is the complement of the basic AND gate.

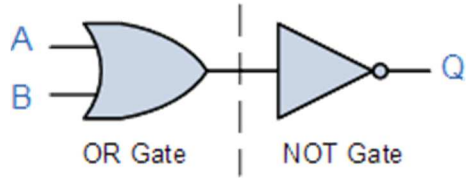
NAND Function Truth Table

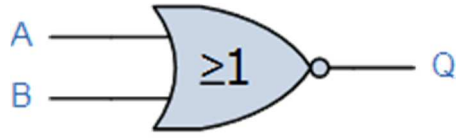
Switch A	Switch B	Output	Description
0	0	1	A and B are both open, lamp ON
0	1	1	A is open and B is closed, lamp ON
1	0	1	A is closed and B is open, lamp ON
1	1	0	A is closed and B is closed, lamp OFF

Boolean Expression (A AND B)	$\overline{A \cdot B}$

The NOR or "Not OR" gate is also a combination of two separate logic functions, Not and OR connected together to form a single logic function which is the same as the OR function except that the output is inverted.

To create a NOR gate, the OR function and the NOT function are connected together in series with its operation given by the Boolean expression as, A + B





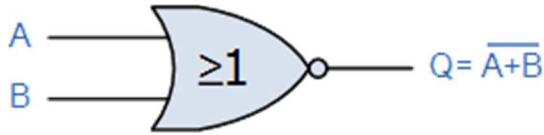
2-input NOR Gate

The **Logic NOR Function** only produces and output when "ALL" of its inputs are not present and in Boolean Algebra terms the output will be TRUE only when all of its inputs are FALSE.

he truth table for the NOR function is the opposite of that for the previous OR function because the NOR gate performs the reverse operation of the OR gate. Then we can see that the NOR gate is the complement of the OR gate.

NOR Function Truth Table

Switch A	Switch B	Output	Description
0	0	1	Both A and B are open, lamp ON
0	1	0	A is open and B is closed, lamp OFF
1	0	0	A is closed and B is open, lamp OFF
1	1	0	A is closed and B is closed, lamp OFF
Boolean Expression (A OR B)			A + B



2-input NOR Gate

Summary

Boolean & DeMorgan's Theorems

1) $X \cdot 0 = 0$
2) $X \cdot 1 = X$
3) $X \cdot X = X$
4) $X \cdot \overline{X} = 0$
5) $X + 0 = X$
6) $X + 1 = 1$
7) $X + X = X$
8) $X + \overline{X} = 1$
9) $\overline{\overline{X}} = X$

10A) $X \cdot Y = Y \cdot X$
10B) $X + Y = Y + X$
— Commutative Law

11A) $X(YZ) = (XY)Z$
11B) $X + (Y + Z) = (X + Y) + Z$
— Associative Law

12A) $X(Y + Z) = XY + XZ$
12B) $(X + Y)(W + Z) = XW + XZ + YW + YZ$
— Distributive Law

13A) $X + \overline{X}Y = X + Y$
13B) $\overline{X} + XY = \overline{X} + Y$
13C) $X + \overline{X}\overline{Y} = X + \overline{Y}$
13D) $\overline{X} + X\overline{Y} = \overline{X} + \overline{Y}$
— Consensus Theorem

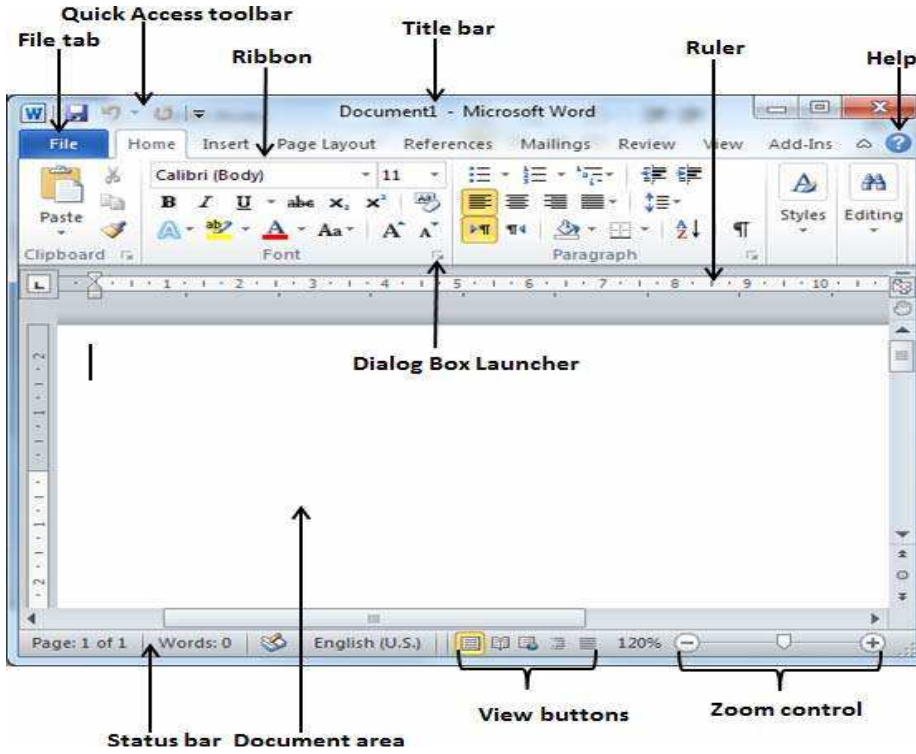
14A) $\overline{X\,Y} = \overline{X} + \overline{Y}$
14B) $\overline{X + Y} = \overline{X}\ \overline{Y}$
— DeMorgan's

6

Chapter IV –

MS- Word

We will understand how to explore Window in Word 2010. Following is the basic window which you get when you start the Word application. Let us understand the various important parts of this window..



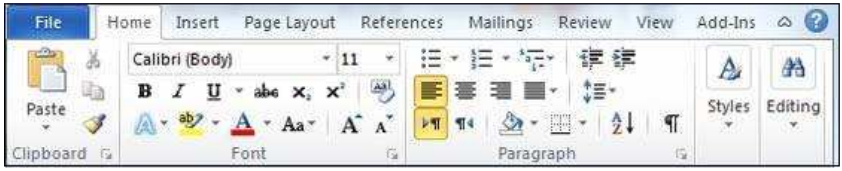
File Tab

The File tab replaces the Office button from Word 2007. You can click it to check the Backstage view. This is where you come when you need to open or save files, create new documents, print a document, and do other file-related operations.

Quick Access Toolbar

This you will find just above the File tab. This is a convenient resting place for the most frequently used commands in Word. You can customize this toolbar based on your comfort.

Ribbon



Ribbon contains commands organized in three components –

Tabs – These appear across the top of the Ribbon and contain groups of related commands. Home, Insert, Page Layout are examples of ribbon tabs.

Groups – They organize related commands; each group name appears below the group on the Ribbon. For example, group of commands related to fonts or group of commands related to alignment, etc.

Commands – Commands appear within each group as mentioned above.

Title bar

This lies in the middle and at the top of the window. Title bar shows the program and document titles.

Rulers

Word has two rulers - a horizontal ruler and a vertical ruler. The horizontal ruler appears just beneath the Ribbon and is used to set margins and tab stops. The vertical ruler appears on the left edge of the Word window and is used to gauge the vertical position of elements on the page.

Help

The Help Icon can be used to get word related help anytime you like. This provides nice tutorial on various subjects related to word.

Zoom Control

Zoom control lets you zoom in for a closer look at your text. The zoom control consists of a slider that you can slide left or right to zoom in or out; you can click the + buttons to increase or decrease the zoom factor.

View Buttons

The group of five buttons located to the left of the Zoom control, near the bottom of the screen, lets you switch through the Word's various document views.

- **Print Layout view** – This displays pages exactly as they will appear when printed.
- **Full Screen Reading view** – This gives a full screen view of the document.
- **Web Layout view** – This shows how a document appears when viewed by a Web browser, such as Internet Explorer.
- **Outline view** – This lets you work with outlines established using Word's standard heading styles.
- **Draft view** – This formats text as it appears on the printed page with a few exceptions. For example, headers and footers aren't shown. Most people prefer this mode.

Document Area

This is the area where you type. The flashing vertical bar is called the insertion point and it represents the location where text will appear when you type.

Status Bar

This displays the document information as well as the insertion point location. From left to right, this bar contains the total number of pages and words in the document, language, etc.

You can configure the status bar by right-clicking anywhere on it and by selecting or deselecting options from the provided list.

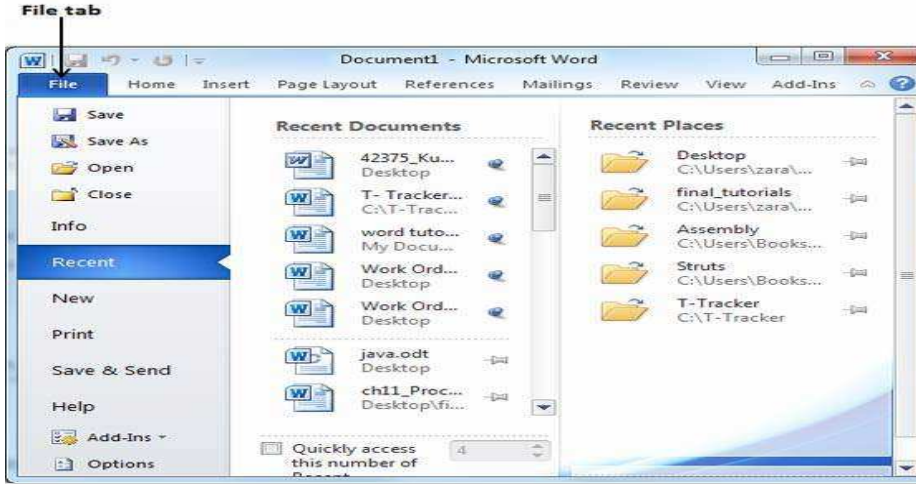
Dialog Box Launcher

This appears as very small arrow in the lower-right corner of many groups on the Ribbon. Clicking this button opens a dialog box or task pane that provides more options about the group.

we will discuss the Backstage View in Word 2010. The Backstage view was introduced in Word 2010. This acts as the central place for

managing your documents. The backstage view helps in creating new documents, saving and opening documents, printing and sharing documents, and so on.

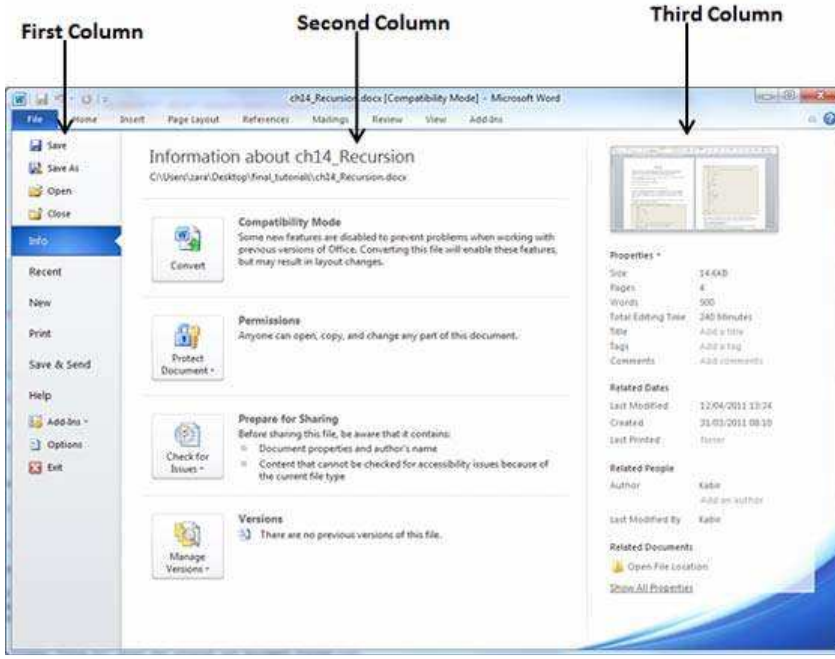
Getting to the Backstage View is easy: Just click the File tab, located in the upper-left corner of the Word Ribbon. If you already do not have any opened document, then you will see a window listing down all the recently opened documents as follows –



If you already have an opened document, then it will display a window showing detail about the opened document as shown below. Backstage view shows three columns when you select most of the available options in the first column.

S.No	Option & Description
1	**Save** If an existing document is opened, it will be saved as is, otherwise it will display a dialogue box asking for the document name.
2	**Save As** A dialogue box will be displayed asking for document name and document type, by default it will save in word 2010 format with extension **.docx**.
3	**Open** This option is used to open an existing word document.
4	**Close** This option is used to close an open document.
5	**Info** This option displays information about the opened document.
6	**Recent** This option lists down all the recently opened documents
7	**New** This option is used to open a new document.
8	**Print** This option is used to print an open document.

9	**Save & Send** This option will save an open document and will display options to send the document using email, etc.
10	**Help** This option is used to get the required help about Word 2010.
11	**Options** This option is used to set various option related to Word 2010.
12	**Exit** Use this option to close the document and exit.



The first column of the backstage view will have following options –

Document Information

When you click the **Info** option available in the first column, it displays the following information in the second column of the backstage view –

- **Compatibility Mode** – If the document is not a native **Word 2007/2010** document, a **Convert** button appears here, enabling you to easily update its format. Otherwise, this category does not appear.
- **Permissions** – You can use this option to protect your word document. You can set a password so that nobody can open your document, or you can lock the document so that nobody can edit your document.

- **Prepare for Sharing** – This section highlights important information you should know about your document before you send it to others, such as a record of the edits you made as you developed the document.

- **Versions** – If the document has been saved several times, you may be able to access the previous versions of it from this section.

Document Properties

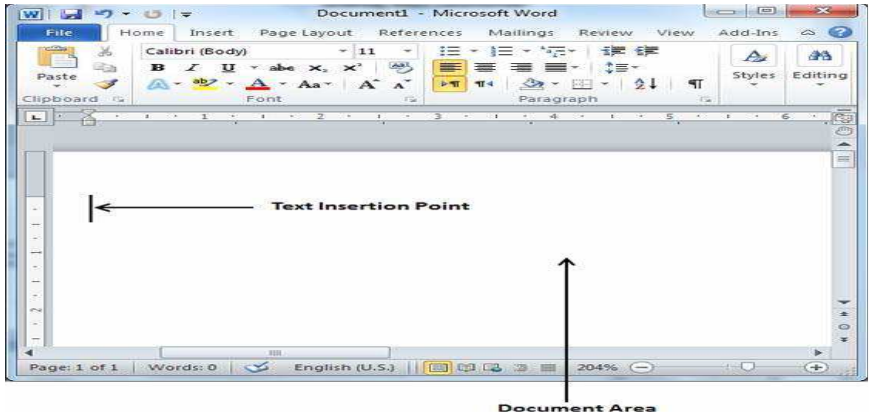
When you click the **Info** option available in the first column, it displays various properties in the third column of the backstage view. These properties include the document size, the number of pages in the document, the total number of words in the document, the name of the author etc.

You can also edit various properties by clicking on the property value and if the property is editable, then it will display a text box where you can add your text like title, tags, comments, Author.

Exit Backstage View

It is simple to exit from the Backstage View. Either click on the File tab or press the Esc button on the keyboard to go back to the working mode of Word.

Let us discuss how to enter text with Microsoft Word 2010. Let us see how easy it is to enter text in a Word document. We assume you know that when you start Word, it displays a new document by default as shown below –



Document area is the area where you type your text. The flashing vertical bar is called the insertion point and it represents the location where the text will appear when you type. keep the cursor at the text insertion point and start typing the text. We typed only two words "Hello Word" as shown below. The text appears to the left of the insertion point as you type –

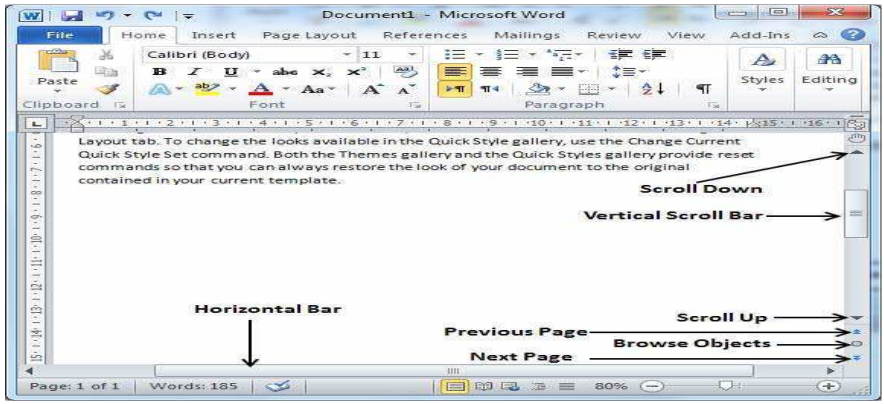
The following are the two important points that will help you while typing –

You do not need to press Enter to start a new line. As the insertion point reaches the end of the line, Word automatically starts a new one. You will need to press Enter, to add a new paragraph.

When you want to add more than one space between words, use the Tab key instead of the spacebar. This way you can properly align text by using the proportional fonts.

Moving with Mouse

You can easily move the insertion point by clicking in your text anywhere on the screen. There may be instances when a document is big and you cannot see a place where you want to move. Here, you will have to use the scroll bars, as shown in the following screenshot –



You can scroll through your document by rolling your mouse wheel, which is equivalent to clicking the up-arrow or down-arrow buttons in the scroll bar.

Moving with Scroll Bars

As shown in the above screenshot, there are two scroll bars: one for moving vertically within the document, and one for moving horizontally. Using the vertical scroll bar, you may –

- Move upward by one line by clicking the upward-pointing scroll arrow.
- Move downward by one line by clicking the downward-pointing scroll arrow.
- Move one next page, using the next page button (footnote).
- Move one previous page, using the previous page button (footnote).

- Use the **Browse Object** button to move through the document, going from one chosen object to the next.

Moving with Keyboard

The following keyboard commands, used for moving around your document, also move the insertion point –

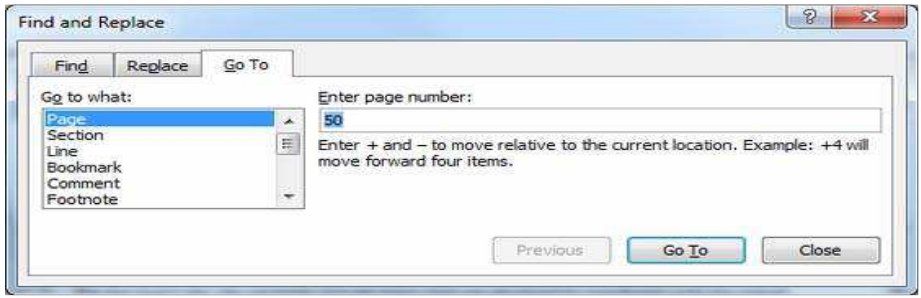
Keystroke	Where the Insertion Point Moves
→	Forward one character
←	Back one character
↑	Up one line
↓	Down one line
PageUp	To the previous screen
PageDown	To the next screen
Home	To the beginning of the current line
End	To the end of the current line

You can move word by word or paragraph by paragraph. You would have to hold down the Ctrl key while pressing an arrow key, which moves the insertion point as described here –

Key Combination	Where the Insertion Point Moves
Ctrl +→	To the next word
Ctrl +←	To the previous word
Ctrl +↑	To the start of the previous paragraph
Ctrl +↓	To the start of the next paragraph
Ctrl + PageUp	To the previous browse object
Ctrl + PageDown	To the next browse object
Ctrl + Home	To the beginning of the document
Ctrl + End	To the end of the document
Shift + F5	To the last place you changed in your document.

Moving with Go To Command

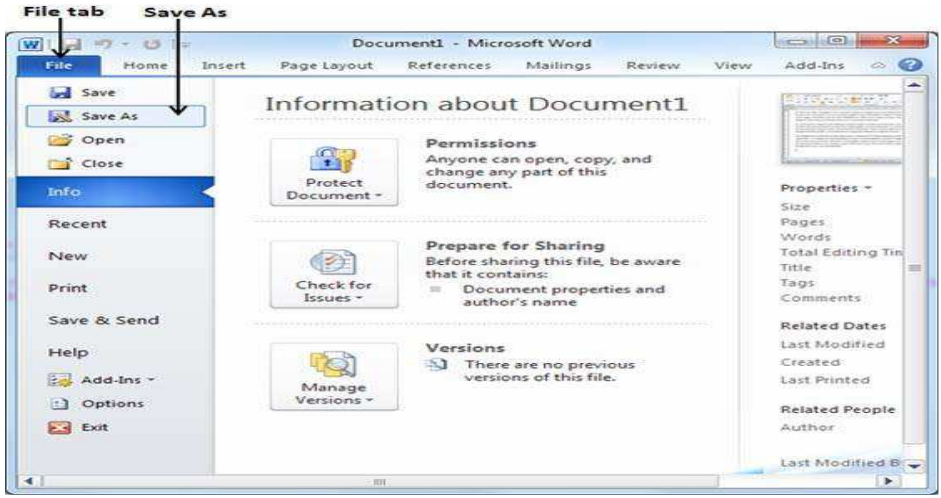
Press the F5 key to use the Go To command. This will display a dialogue box where you will have various options to reach to a particular page. Normally, we use the page number, the line number or the section number to go directly to a particular page and finally press the Go To button.



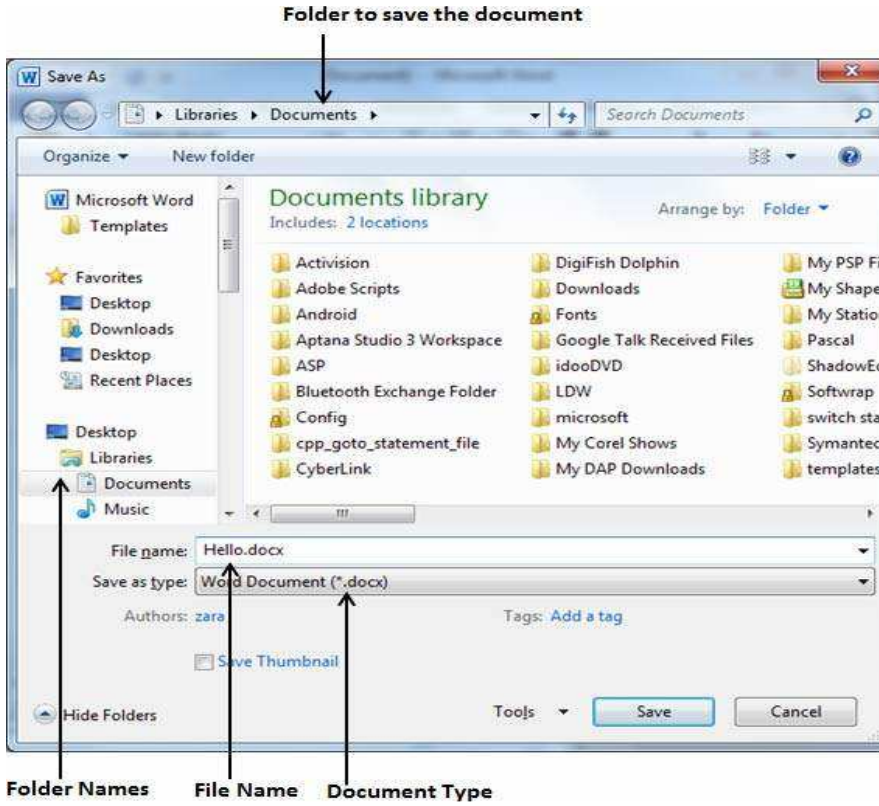
Saving New Document

Once you are done with typing in your new Word document, it is time to save your document to avoid losing work you have done on a Word document. Following are the steps to save an edited Word document –

Step 1 – Click the File tab and select the Save As option.



Step 2 – Select a folder where you will like to save the document, Enter the file name which you want to give to your document and Select the Save As option, by default it is the .docx format.



Step 3 – Finally, click on the Save button and your document will be saved with the entered name in the selected folder.

Saving New Changes

There may be an instance when you open an existing document and edit it partially or completely, or an instance where you may like to save the changes in between editing of the document. If you want to save this document with the same name, then you can use either of the following simple options –

- Just press the **Ctrl + S** keys to save the changes.

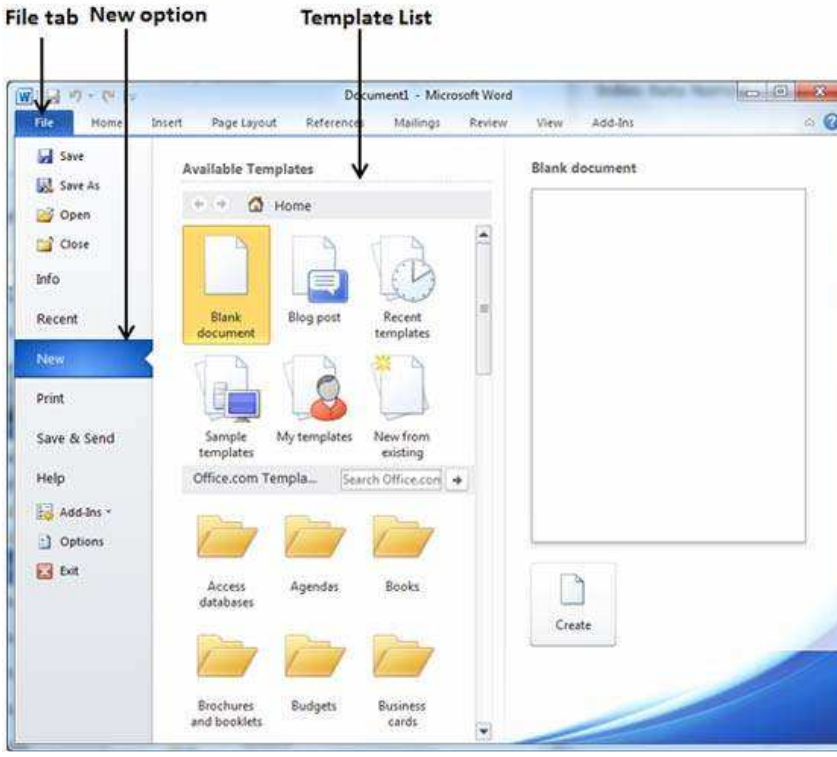
- Optionally you can click on the floppy icon available at the top left corner and just above the **File tab**. This option will also help you save the changes.
- You can also use the third method to save the changes, which is the **Save** option available just above the **Save As** option as shown in the above screenshot.

If your document is new and it was never saved so far, then with either of the three options, Word will display a dialogue box to let you select a folder, and enter the document name as explained in case of saving new document.

pening New Document

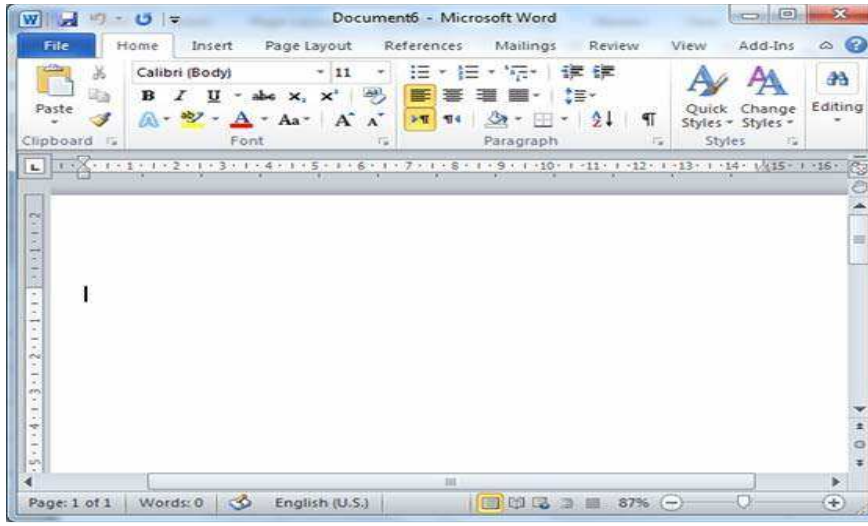
A new, blank document always opens when you start Microsoft Word. Suppose you want to start another new document while you are working on another document, or you closed an already opened document and want to start a new document. Here are the steps to open a new document –

Step 1 – Click the File tab and select the New option.



Step 2 – When you select the New option from the first column, it will display a list of templates in the second column. Double-click on the Blank document; this is the first option in the template list. We will discuss the other templates available in the list in the following chapters.

You should have your blank document as shown below. The document is now ready for you to start typing your text.

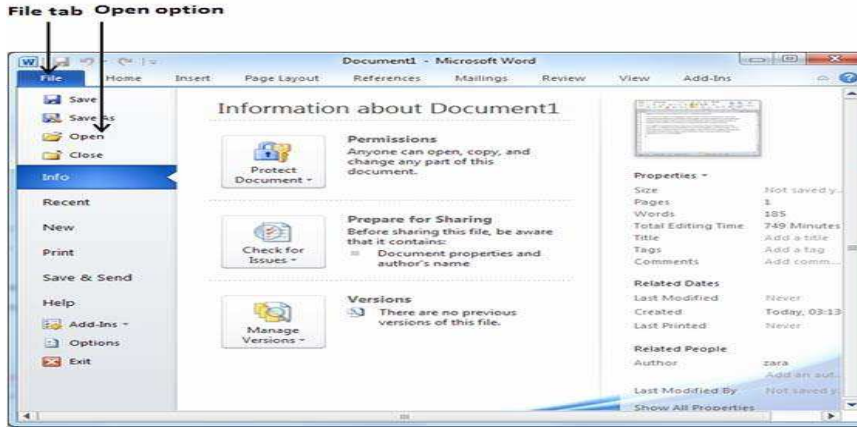


You can use a shortcut to open a blank document anytime. Try using the Ctrl + N keys and you will see a new blank document similar to the one in the above screenshot.

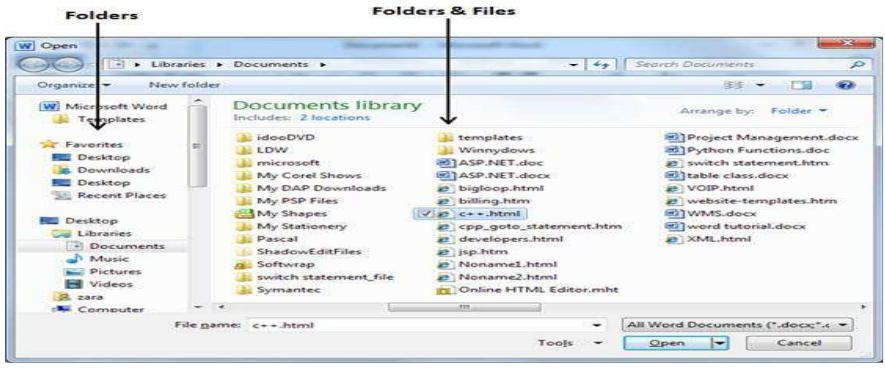
Opening Existing Document

There may be a situation when you open an existing document and edit it partially or completely. Follow the steps given below to open an existing document –

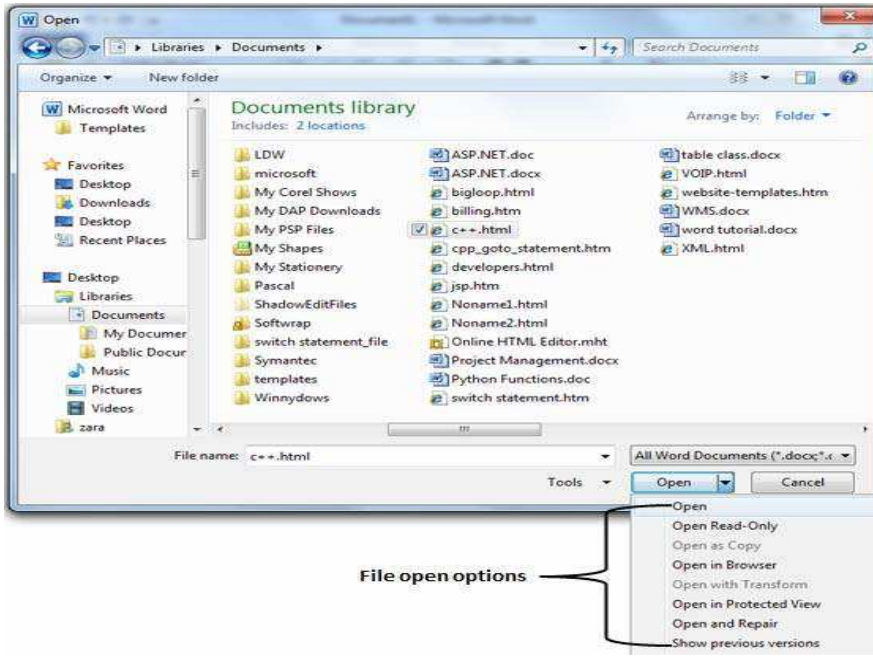
Step 1 – Click the File tab and select the Open option.



Step 2 – This will display the following file Open dialog box. This lets you navigate through different folders and files, and also lets you select a file which you want to open.



Step 3 – Finally, locate and select a file which you want to open and click the small triangle available on the Open button to open the file. You will have different options to open the file, but simply use the Open option.

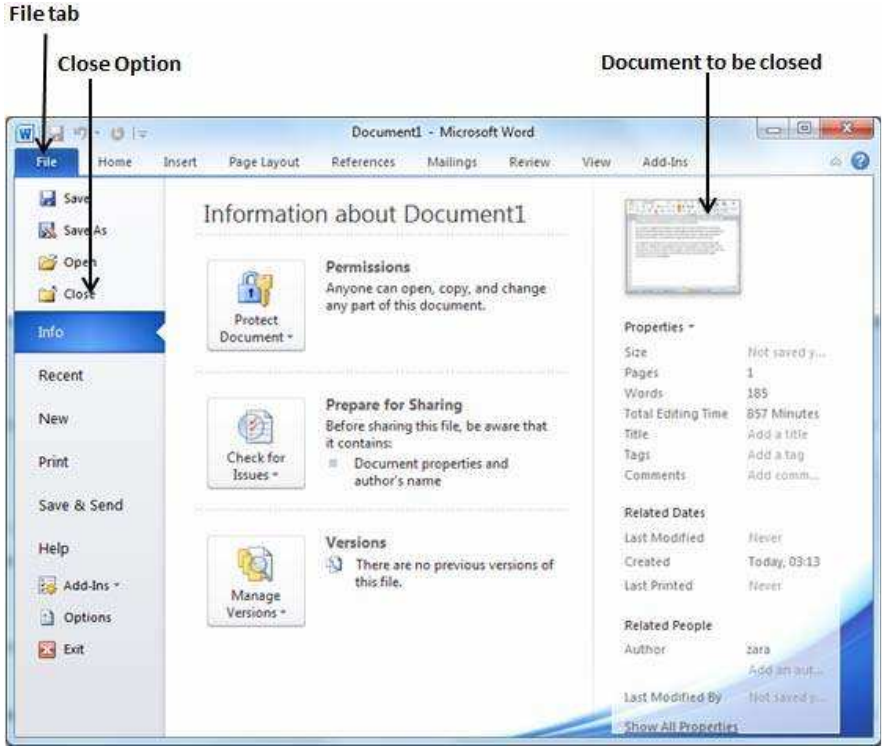


This will open your selected file. You can use the Open Read-Only option if you are willing just to read the file and you have no intention to modify, i.e., edit the file. Other options can be used for advanced usage.

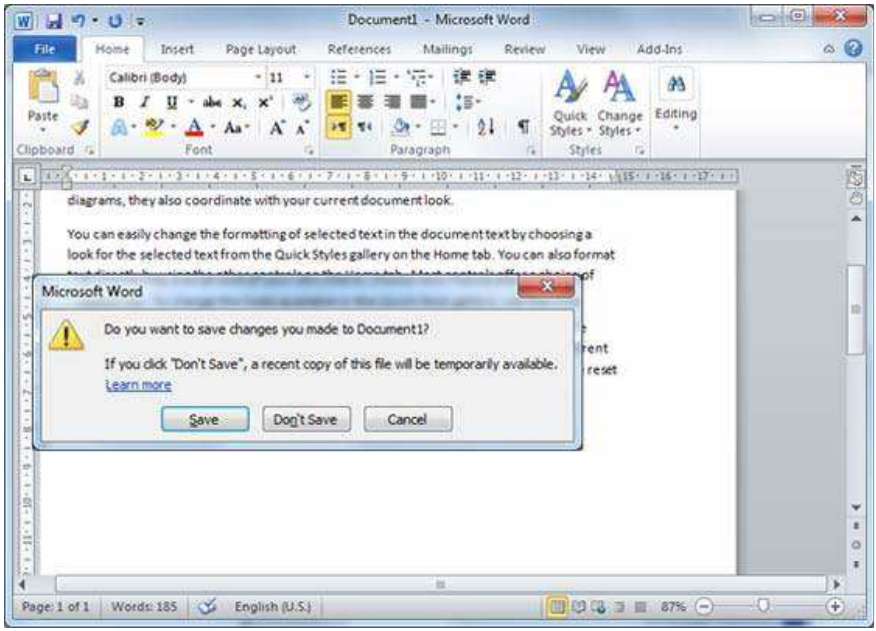
CLOSING A DOCUMENT

Let us understand how to close a document in Word 2010. When you finish working with a document, you will proceed to close the document. Closing a document removes it from your computer screen and if you had other documents open, Word displays the last document you used otherwise, you see a blank Word window. Here are simple steps to close an opened document –

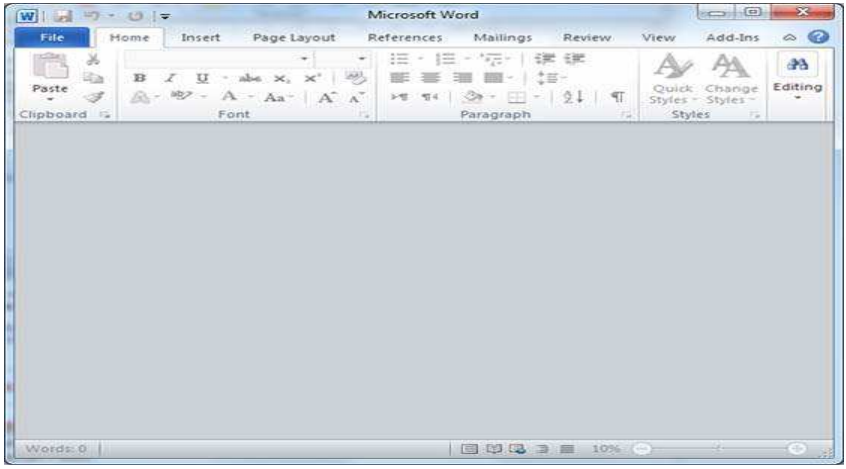
Step 1 – Click the File tab and select the Close option.



Step 2 – When you select the Close option and if the document is not saved before closing, it will display the following Warning box asking whether the document should be saved or not.



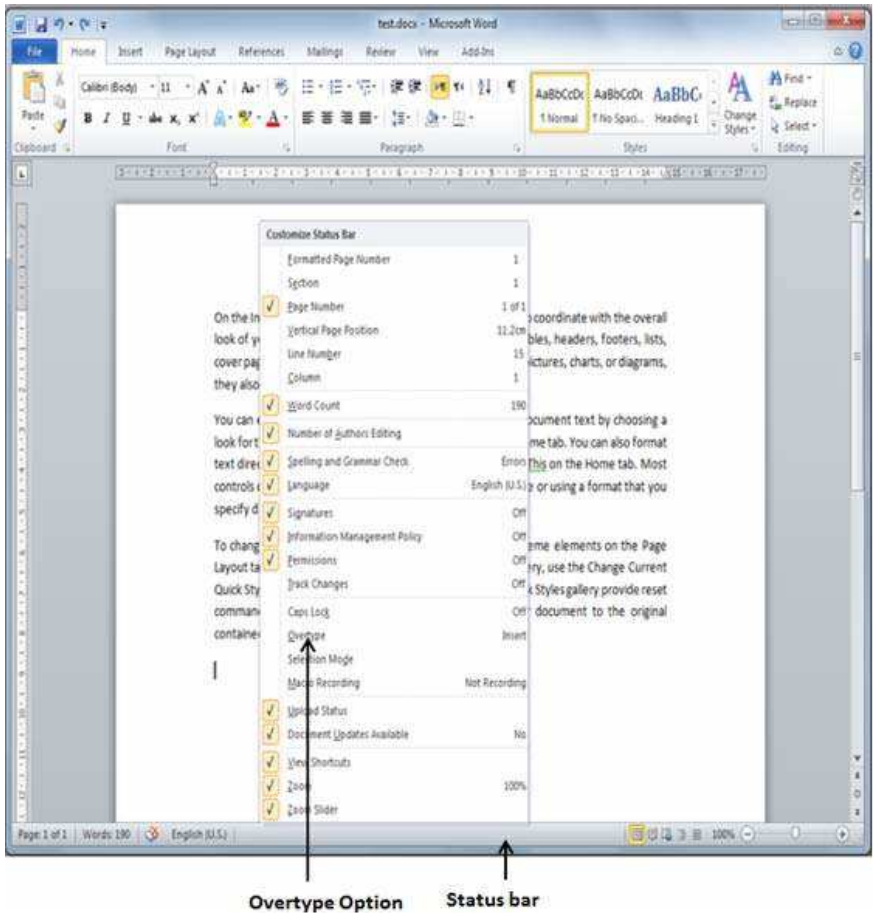
Step 3 – To save the changes, click Save, otherwise click Don't Save. To go back to the document, click Cancel. This will close the document and if you have other documents open, Word displays the last document you used, otherwise, you see a blank Word window as shown below –



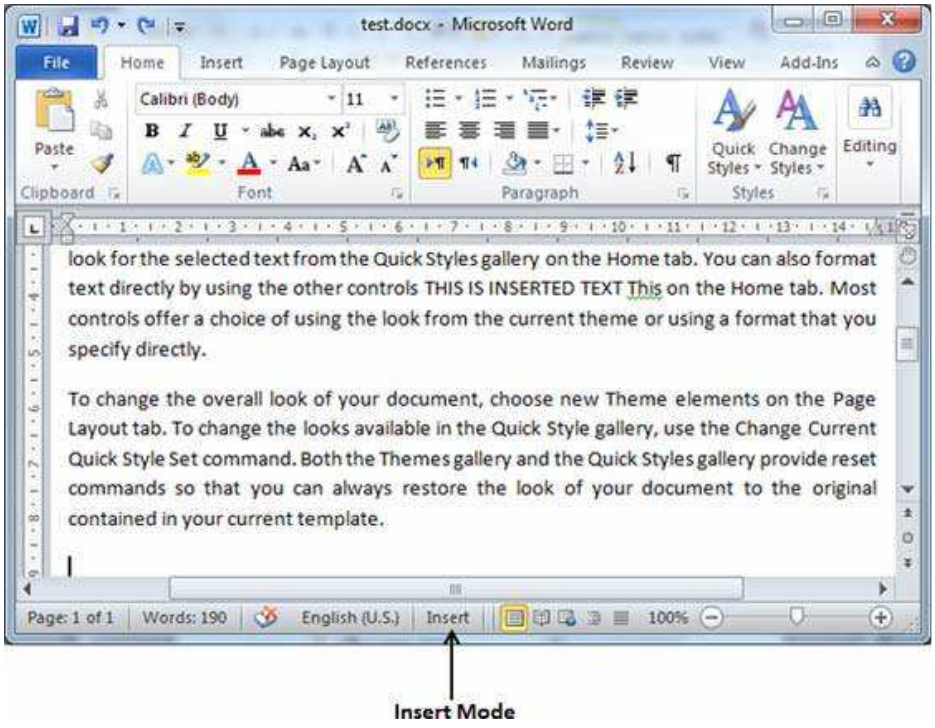
Insert and Replace Text

In the Insertion mode, text will be added into the existing content but same time it will over write all the content which comes in its way.

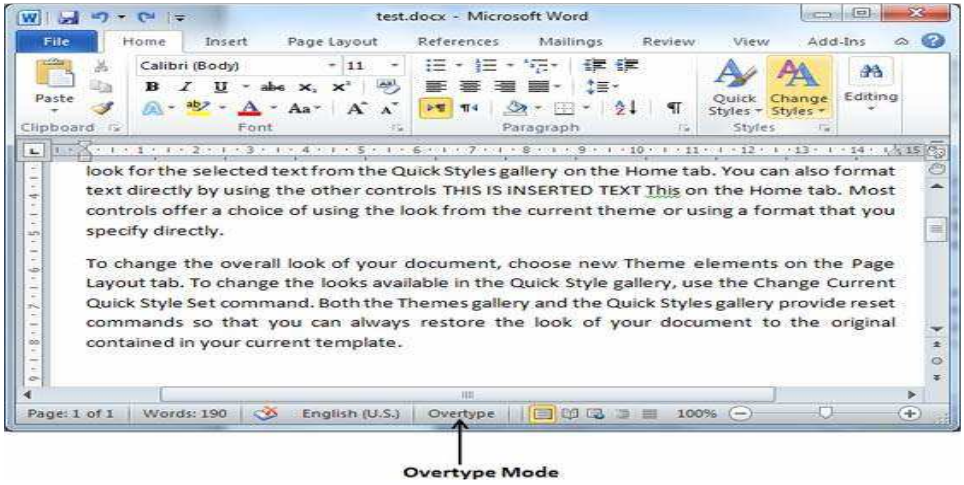
Step 1 – Right-click the status bar and select the **Overtype** option from the displayed menu.



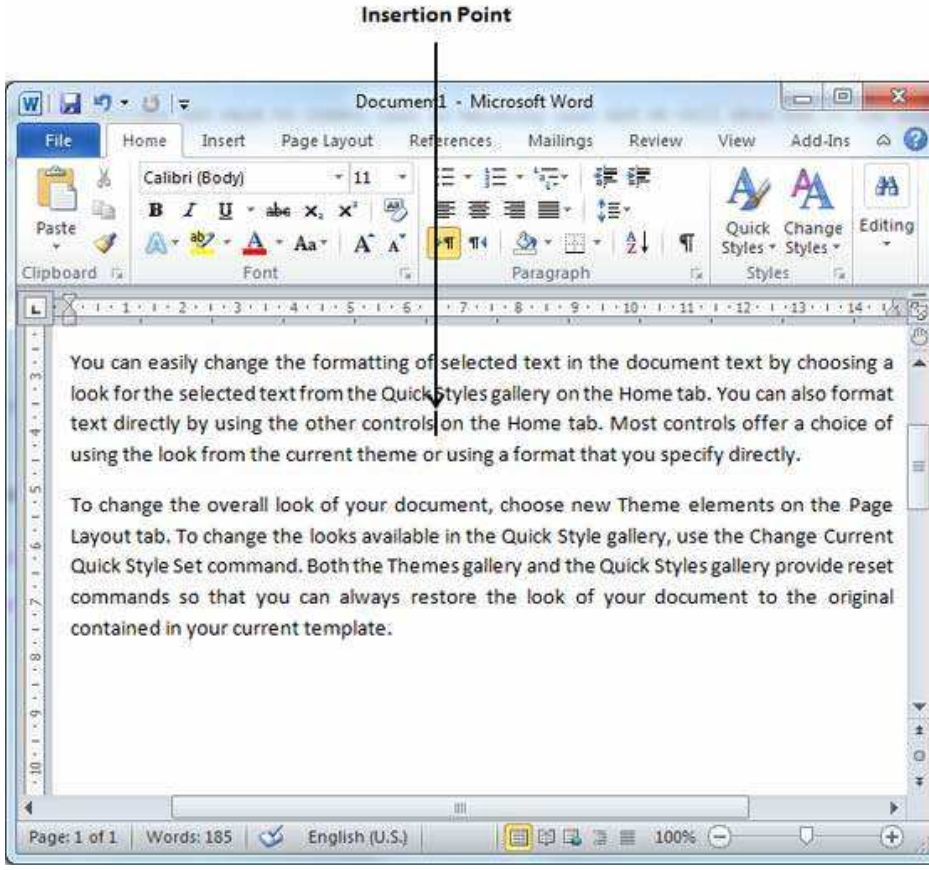
When you select the **Overtype** option, the status bar will show the **insert** mode as shown below –



Step 2 – Click on the **Insert** text available at the status bar and it will switch to the **Overtype** mode as shown below –

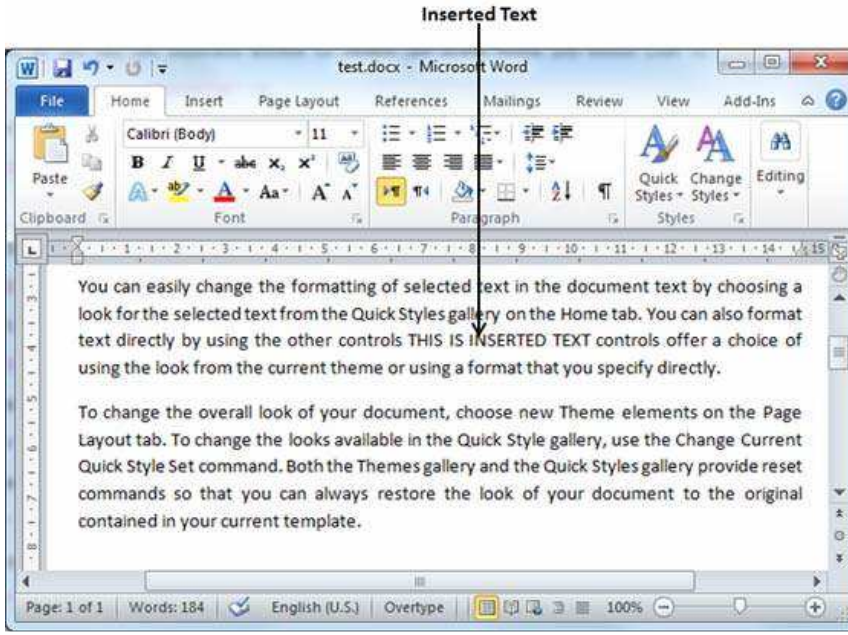


Step 3 – Now click the location where the text needs to be inserted or you can use the keyboard arrows to locate the place where the text needs to be inserted.



Step 4 – Start typing the text that needs to be inserted. Word will replace the existing text with the newly typed text without moving the position of the exiting test.

S.No	Component & Selection Method
1	**Selecting text between two points** Click at the start of the block of text, hold down **Shift**, and click at the end of the block.
2	**Selecting a single word** Double-click anywhere on the word you want to select.
3	**Selecting a paragraph** Triple-click anywhere on the paragraph you want to select.
4	**Selecting a sentence** Hold down the **Ctrl** key and click anywhere in the sentence you want to select.
5	**Selecting a column of text** Hold down Alt, click and hold the mouse button, and drag over the column you want to select.



Note – Microsoft Word 2010 disabled the functionality of the **Insert** key and it does nothing, so you will have to follow-up with the above mentioned procedure to turn-on or turn-off the Insert mode.

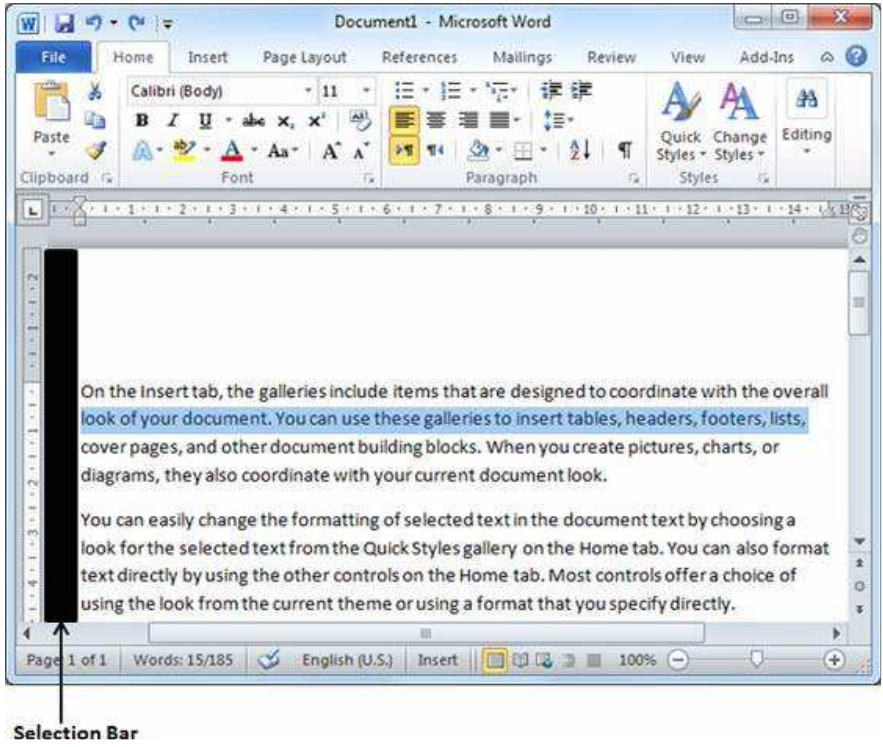
we will discuss how to select text in Word 2010. Selecting a text is one of the most important skills required while editing a word document. You can perform various operations on a selected text; you can delete the selected text, copy it, move it, apply formatting to it, change its capitalization, etc.

The most common method of selecting a text is to click and drag the mouse over the text you want to select. Following table lists down a few other simple methods that will help you in selecting text in different scenarios –

Note that only one part of the document can be in the selected state. If you have one portion of the document in selected state and as soon as you try to select any other part of the document, previous part will automatically be de-selected.

Using the Selection Bar

The black shaded area in the following screen shot is called the **selection bar**. When you bring your cursor in this area, it turns into a rightward-pointing arrow.



You can use the selection bar to select the various components of a document as described in the following table –

S.No	Component & Selection Method
1	**Selecting a line** Bring your mouse in the selection bar area and click in front of

	the line you want to select.
2	**Selecting a paragraph** Bring your mouse in the selection bar area and double click in front of the paragraph you want to select.
3	**Selecting the document** Bring your mouse in the selection bar area and triple-click.

Using the Keyboard

Keyboard provides very good support when you want to select various components of the document as described in the following table –

Key & Selection Method Selecting Text
Ctrl + A Press **Ctrl + A** keys to select the entire document.
Shift Keep pressing the **Shift** key and use any of the arrow keys to select the portion of text.
F8 Press **F8** and then use any of the arrows keys to select the portion of text.

Ctrl + Shift + F8

Press **Ctrl + Shift + F8** and then use any of the arrows keys to select column of the text.

Using Backspace & Delete Keys

The most basic deletion technique is to delete characters one at a time by pressing either the backspace key or the delete key. Following table describes how you can delete single character or a whole word by using either of these two keys –

S.No	Keys & Deletion Methods
1	**Backspace** Keep the insertion point just after the character you want to delete and press the **Backspace** key. Word deletes the character immediately to the left of the insertion point.
2	**Ctrl + Backspace** Keep the insertion point just after the word you want to delete and press **Ctrl + Backspace** key. Word deletes the whole word immediately to the left of the insertion point.
3	**Delete** Keep the insertion point just before the character you want to delete and press the **Delete** key. Word deletes the character immediately to the right of the insertion point.

4	**Ctrl + Delete** Keep the insertion point just before the word you want to delete and press **Ctrl + Delete** key. Word deletes the word immediately to the right of the insertion point.

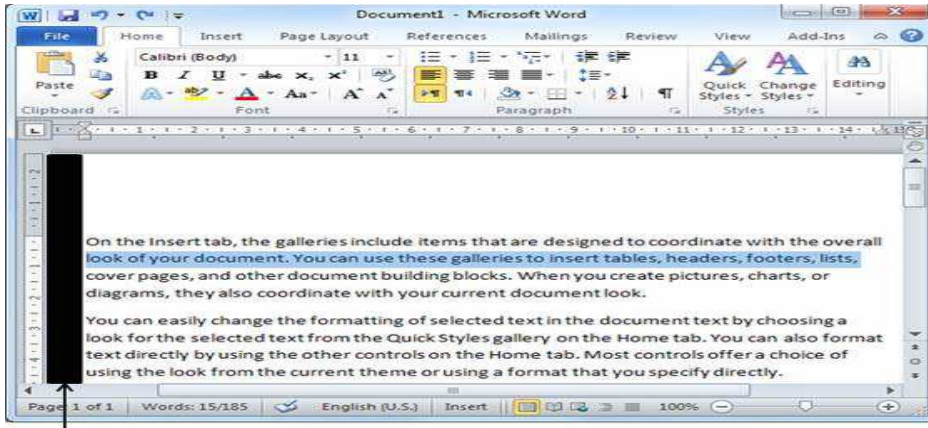
Using Selection Method

You have learnt how to select various parts of a Word document. You can make use of that learning to delete those selected parts as described in the following table –

S.No	Component Selection & Delete Methods
1	**Deleting text between two points** Click at the start of the block of text, hold down the **Shift** key, and click at the end of the block to select the portion of text and finally press either the **Backspace** key or the **Delete** key.
2	**Deleting a single word** Double-click anywhere on the word you want to delete and finally press either the **Backspace** key or the **Delete** key.
3	**Deleting a paragraph** Triple-click anywhere on the paragraph you want to delete and finally press either the **Backspace** key or the **Delete** key.

4	**Deleting a sentence** Hold down the **Ctrl** key and click anywhere in the sentence you want to delete and finally press either the **Backspace** or the **Delete** key.
5	**Deleting a column of text** Hold down the **Alt** key, click and hold the mouse button, and drag over the column you want to delete and finally press either the **Backspace** key or the **Delete** key.
6	**Deleting a line** Bring your mouse in the selection bar area and click in front of the line you want to delete and finally press either the **Backspace** key or the **Delete** key.
7	**Deleting entire document content** Press **Ctrl + A** keys to delete the entire document and finally press either the **Backspace** key or the **Delete** key.

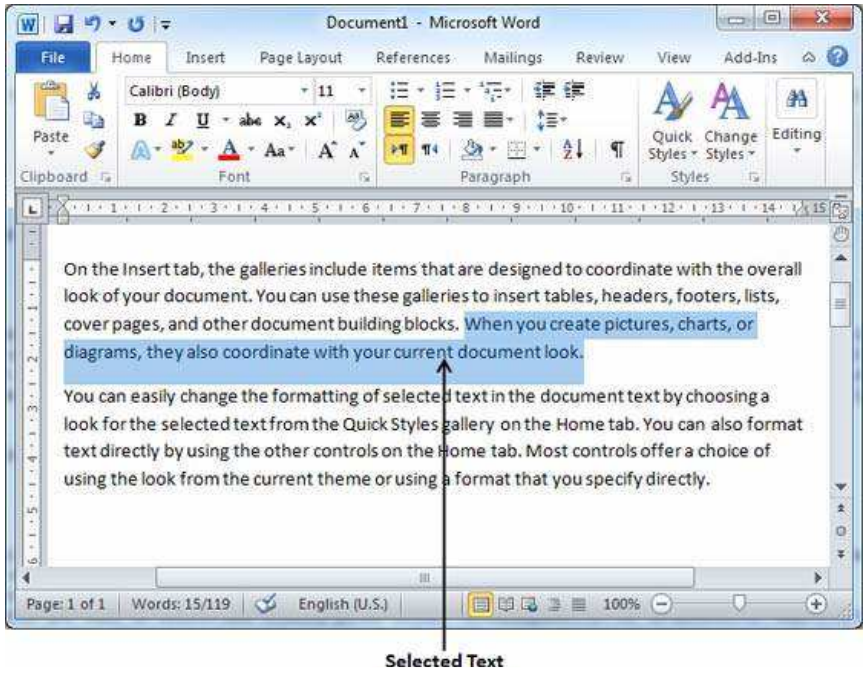
Note – The black shaded area in the following screen shot is called the **selection bar**. When you bring your cursor in this area, it turns into a rightward-pointing arrow.



Copy & Paste Operation

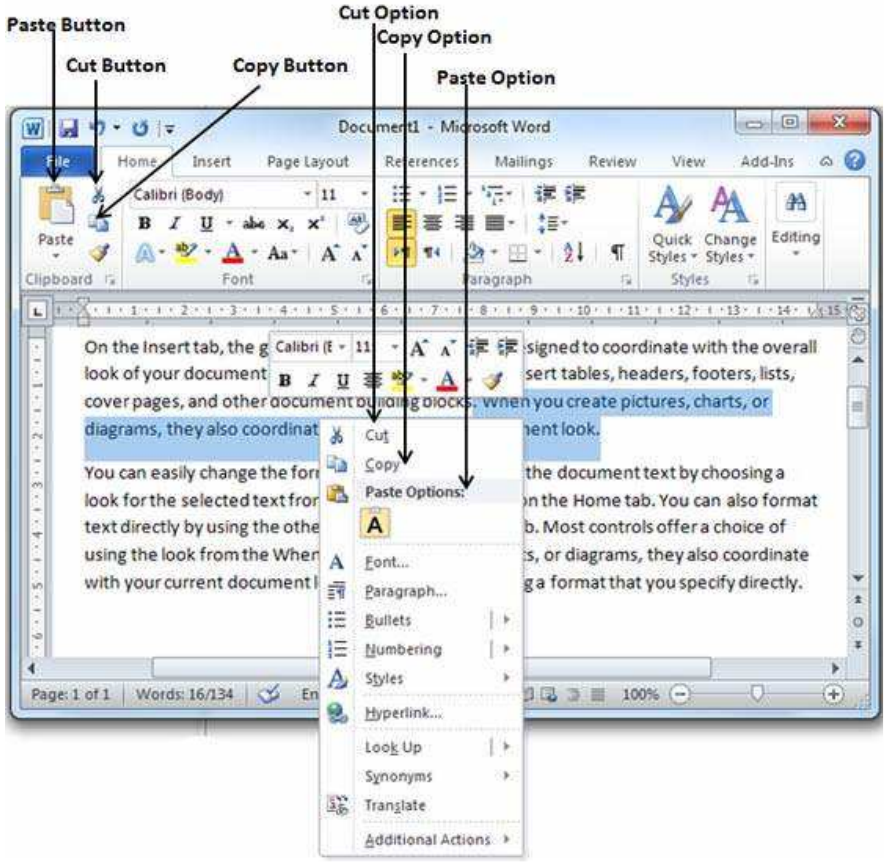
The **Copy** operation will just copy the content from its original place and create a duplicate copy of the content at the desired location without deleting the text from it's the original location. Following is the procedure to copy the content in word –

Step 1 – Select a portion of the text using any of the text selection methods.



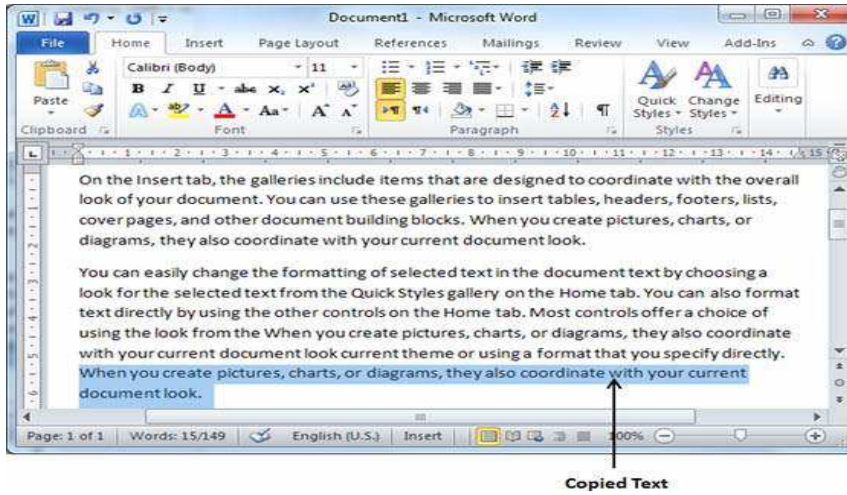
Step 2 – You have various options available to copy the selected text in clipboard. You can make use of any one of the options –

- **Using Right-Click** – When you right-click on the selected text, it will display the **copy** option, click this option to copy the selected content in clipboard.
- **Using Ribbon Copy Button** – After selecting text, you can use the copy button available at the ribbon to copy the selected content in clipboard.
- **Using Ctrl + c Keys** – After selecting a text, just press **Ctrl + c** keys to copy the selected content in clipboard.



Step 3 – Finally click at the place where you want to copy the selected text and use either of these two simple options –

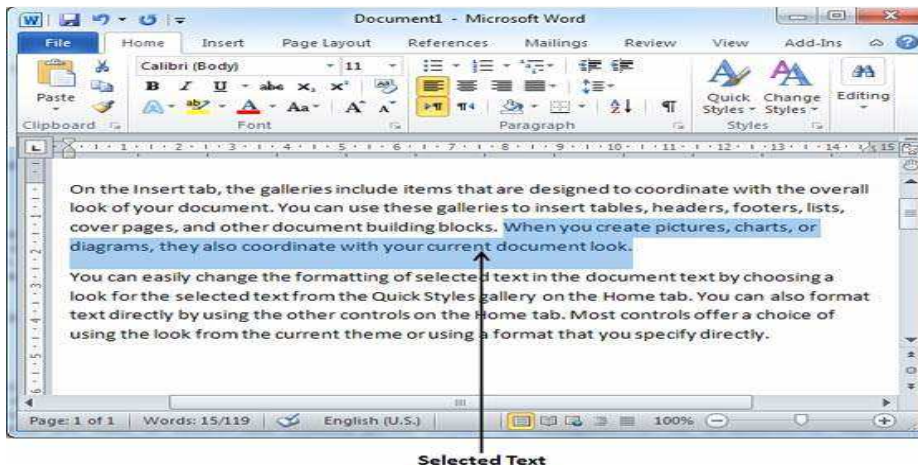
- **Using Ribbon Paste Button** – Just click the Paste button available at the ribbon to paste the copied content at the desired location.
- **Using Ctrl + v Keys** – This is simplest way of pasting the content. Just press **Ctrl + v** keys to paste the content at the new location.



Note – You can repeat the **Paste** operation as many times as you like to paste the same content.

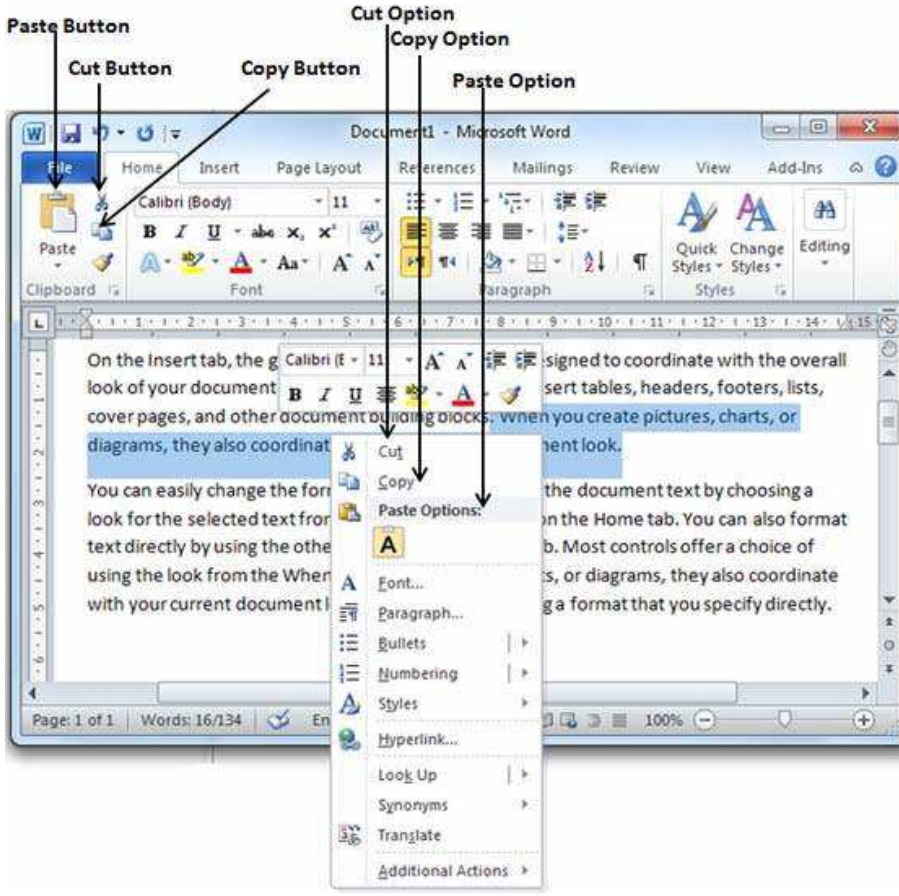
Cut & Paste Operation

The Cut operation will cut the content from its original place and move the content from its original location to a new desired location. Following is the procedure to move the content in word **Step 1** – Select a portion of the text using any of the text selection methods.



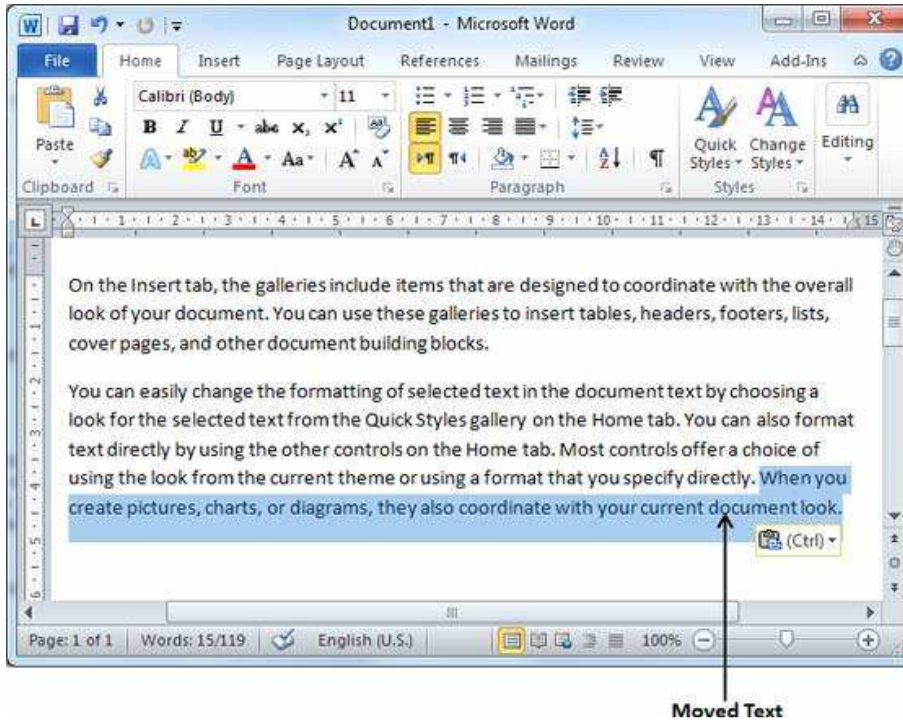
Step 2 – Now, you have various options available to cut the selected text and put it in the clipboard. You can make use of one of the options –

- **Using Right-Click** – If right-click on the selected portion of text, it will display **cut** option, just click this option to cut the selected content and keep it in clipboard.
- **Using Ribbon Cut Button** – After selecting a portion of text, you can use cut button available at the ribbon to cut the selected content and keep it in clipboard.
- **Using Ctrl + x Keys** – After selecting a portion of text, just press **Ctrl + x** keys to cut the selected content and keep it in clipboard.



Step 3 – Finally, click at the place where you want to move the selected text and use either of these two simple options –

- **Using Ribbon Paste Button** – Just click the **Paste** button available at the ribbon to paste the content at the new location.
- **Using Ctrl + v Keys** – This is simplest way of pasting the content. Just press **Ctrl + v** keys to paste the content at the new location.



Note – You can repeat the **Paste** operation as many times as you like to paste the same content.

Copy, Cut & Paste in different documents

You can use the same procedure that we discussed above to **copy and paste** or **cut and paste** content from one document to another document. This is very simple, just copy or cut the desired content from one document and go into another document where you want to paste the content and use mentioned step to paste the content.

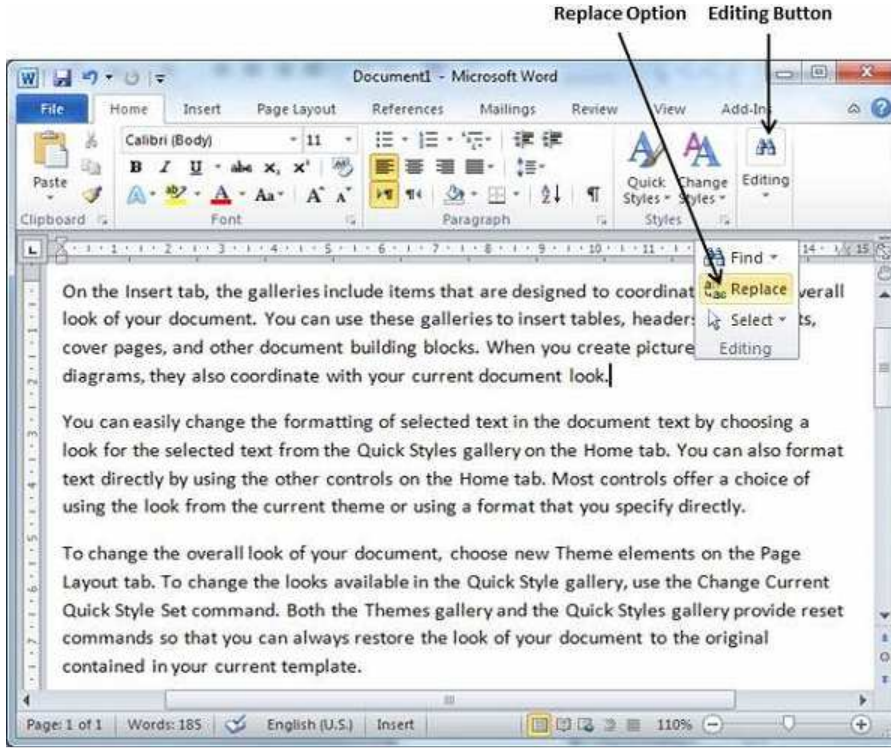
You can use the **Alt + Tab** keys to switch through the different documents and select the desired destination document.

Find and Replace

Find & Replace Operation

We assume you are an expert in searching a word or phrase in a word document as explained above. This section will teach you how you can replace an existing word in your document. Following are the simple steps –

Step 1 – Click the **Replace option** in the **Editing group** on the Home tab or press **Ctrl + H** to launch the **Find and Replace** dialog box shown in Step 2 –



Step 2 – Type a word which you want to search. You can also replace the word using the **Find and Replace** dialog box –

Step 3 – Click the **Replace** button available on the **Find and Replace** dialog box and you will see the first occurrence of the searched word would be replaced with the replace with word. Clicking again on **Replace** button would replace next occurrence of the searched word. If you will click **Replace All** button then it would replace all the found words in one go. You can also use **Find Next** button just to search the next occurence and later you can use **Replace** button to replace the found word.

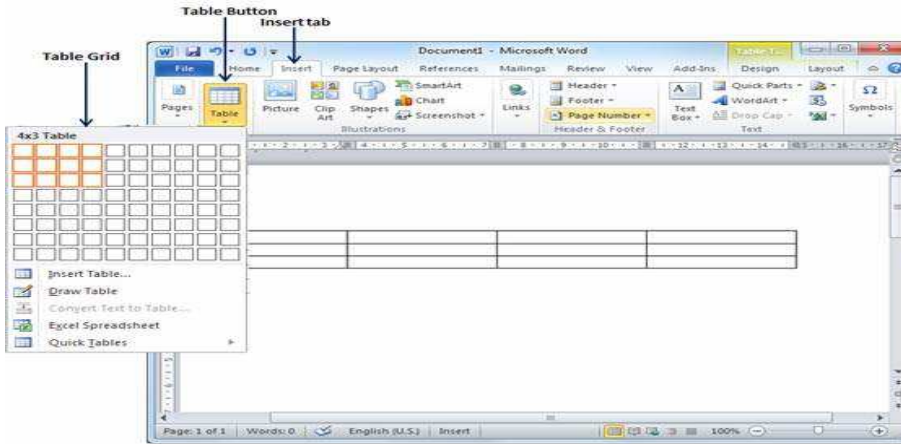
Step 4 – You can use More >> button available on the dialog box to use more options and to make your search more specific like case sensitive search or searching for whole word only etc.

Step 5 – Finally, if you are done with the Find and Replace operation, you can click the Close (X) or Cancel button of the dialog box to close the box.

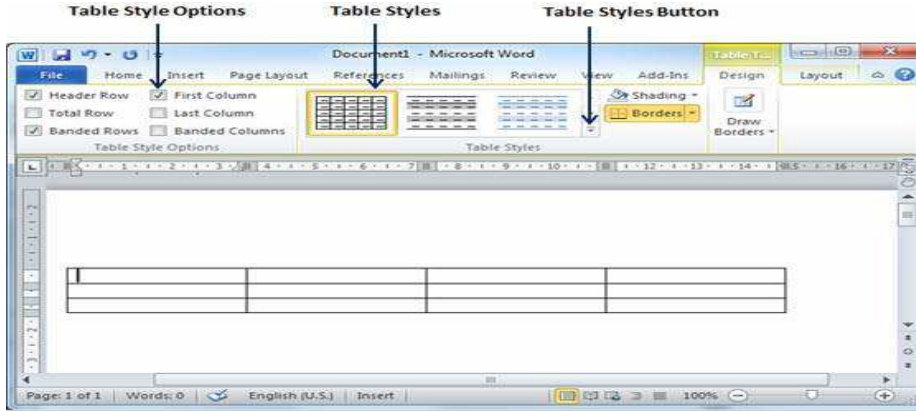
Create a Table

The following steps will help you understand how to create a table in a Word document.

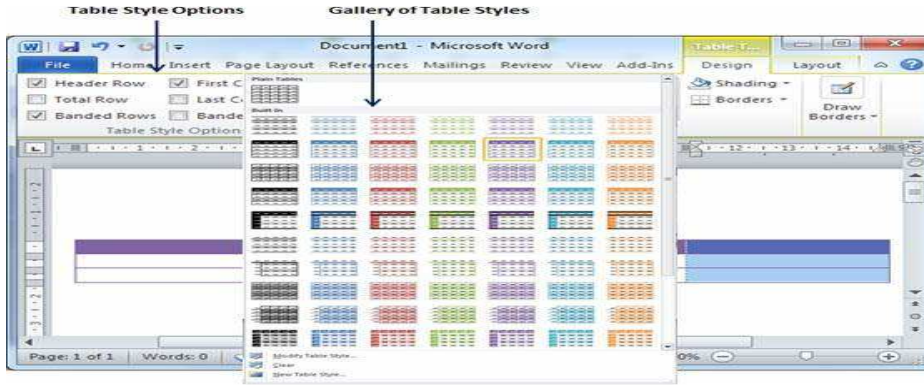
Step 1 – Click the Insert tab followed by the Table button. This will display a simple grid as shown below. When you move your mouse over the grid cells, it makes a table in the table that appears in the document. You can make your table having the desired number of rows and columns.



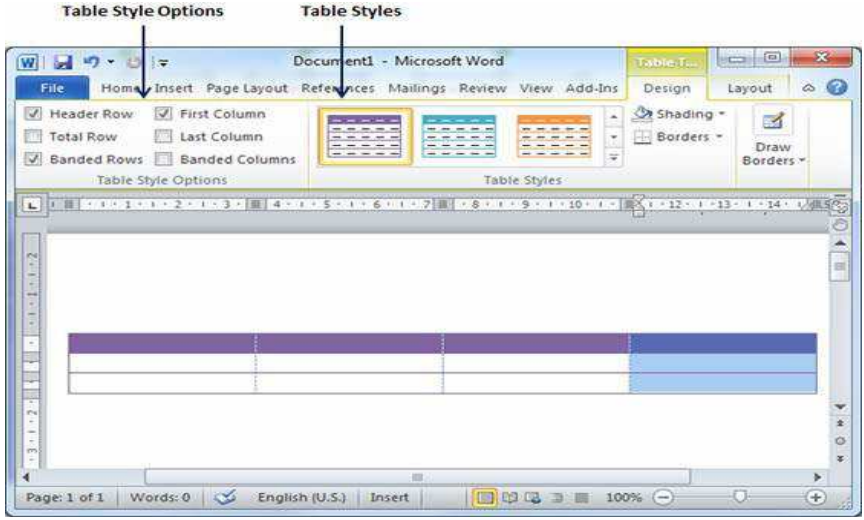
Step 2 – Click the square representing the lower-right corner of your table, which will create an actual table in your document and Word goes in the table design mode. The table design mode has many options to work with as shown below.



Step 3 – This is an optional step that can be worked out if you want to have a fancy table. Click the **Table Styles button** to display a gallery of table styles. When you move your mouse over any of the styles, it shows real time preview of your actual table.



Step 4 – To select any of the styles, just click the built-in table style and you will see that the selected style has been applied on your table.



Delete a Table

Following are the simple steps to delete an existing table from a word document.

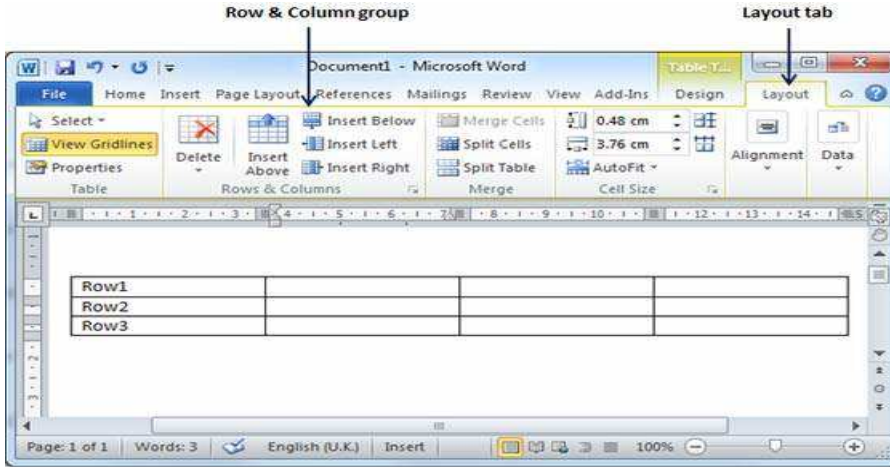
Step 1 – Click anywhere in the table you want to delete.

Step 2 – Click the **Layout tab**, and click the **Delete Table option** under the **Delete Table Button** to delete the complete table from the document along with its content.

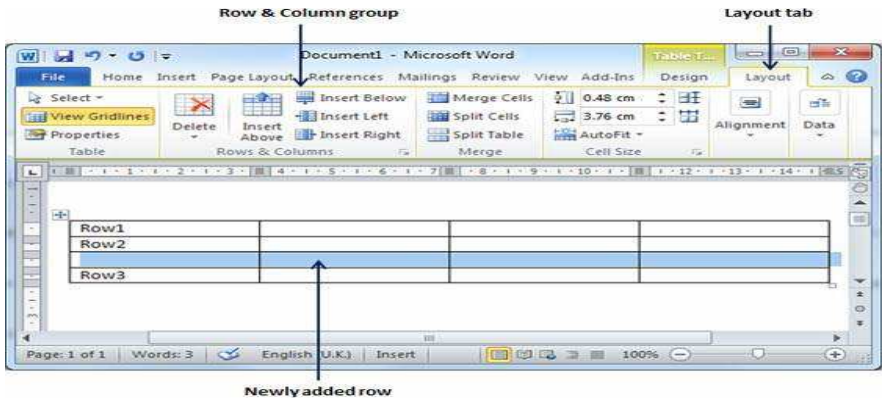
Add a Row

Following are the simple steps to add rows in a table of a word document.

Step 1 – Click a row where you want to add an additional row and then click the **Layout tab**; it will show the following screen.



Step 2 – Now use the **Row & Column group** of buttons to add any row below or above to the selected row. If you click the **Insert Below** button, it will add a row just below the selected row as follows.

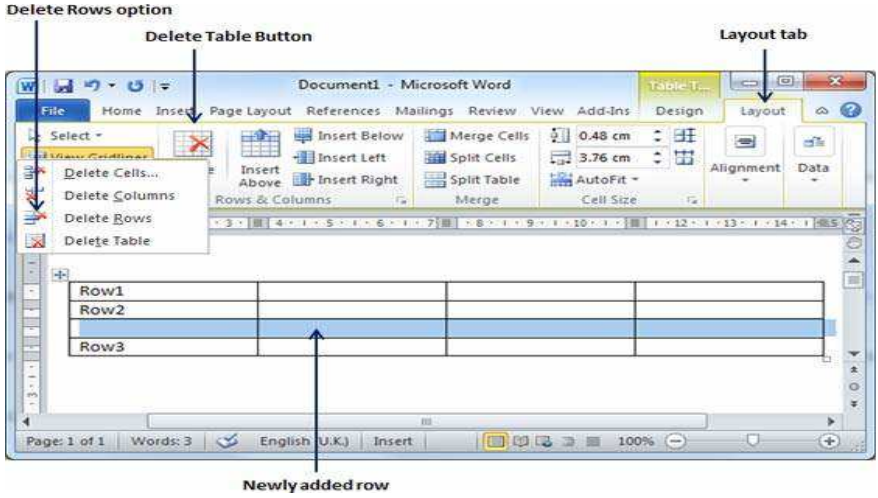


If you click the **Insert Above** button, it will add a row just above the selected row.

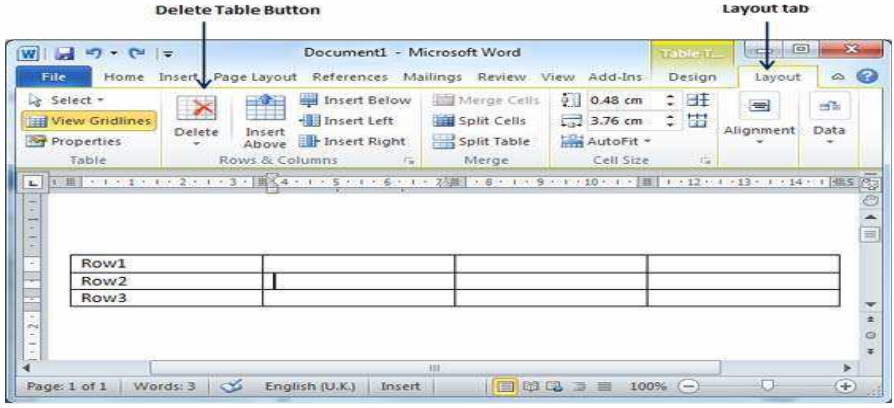
Delete a Row

The following steps will help you delete rows from a table of a Word document.

Step 1 – Click a row which you want to delete from the table and then click the **Layout tab**; it will show the following screen.



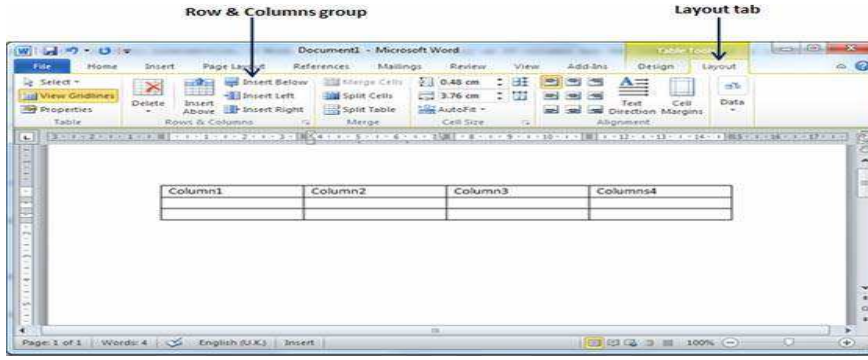
Step 2 – Click the **Layout tab**, and then click the **Delete Rows** option under the **Delete Table Button** to delete the selected row.



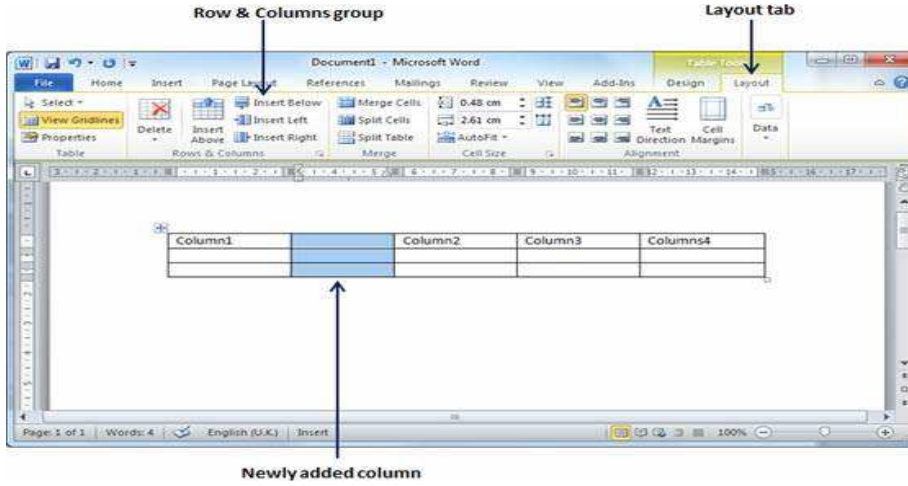
Add a Column

The following steps will help you add columns in a table of a Word document.

Step 1 – Click a column where you want to add an additional column and then click the **Layout tab**; it will show the following screen.



Step 2 – Now use the **Row & Column** group of buttons to add any column to the left or right of the selected column. If you click the **Insert Left** button, it will add a column just left to the selected column as follows.

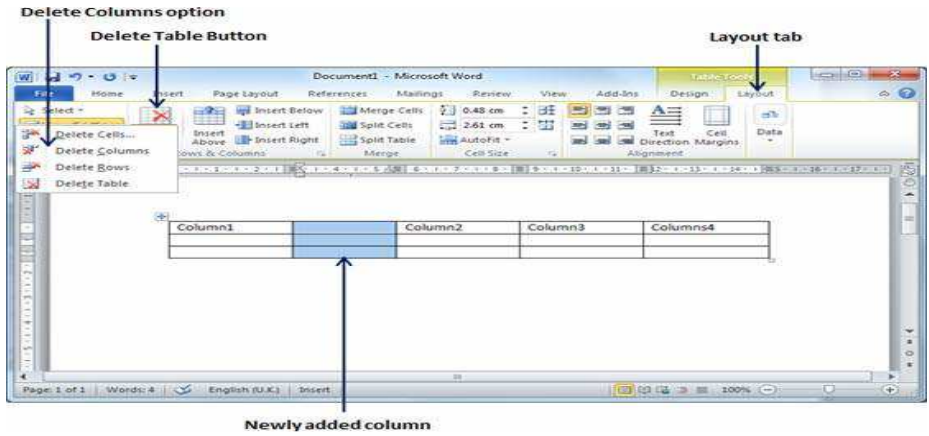


If you click the **Insert Right** button, it will add a column just next to the selected column.

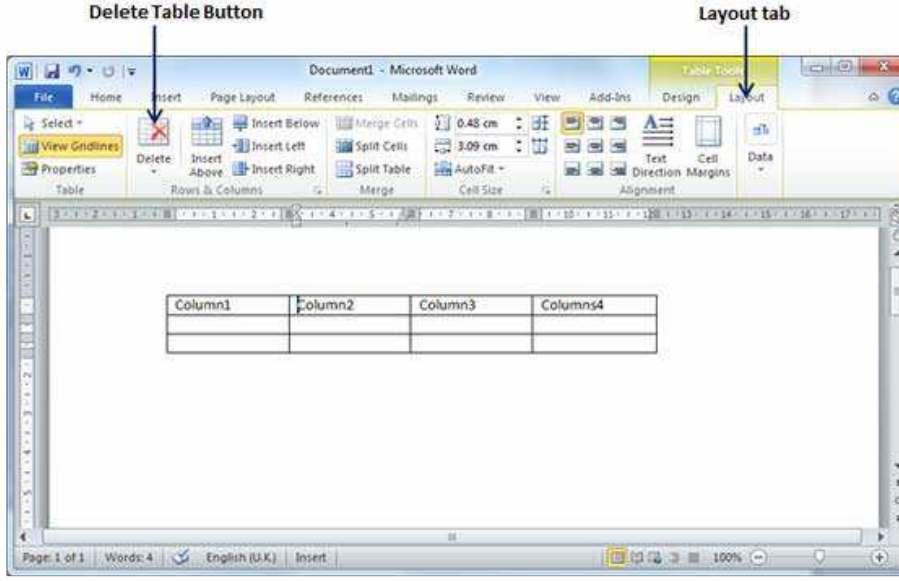
Delete a Column

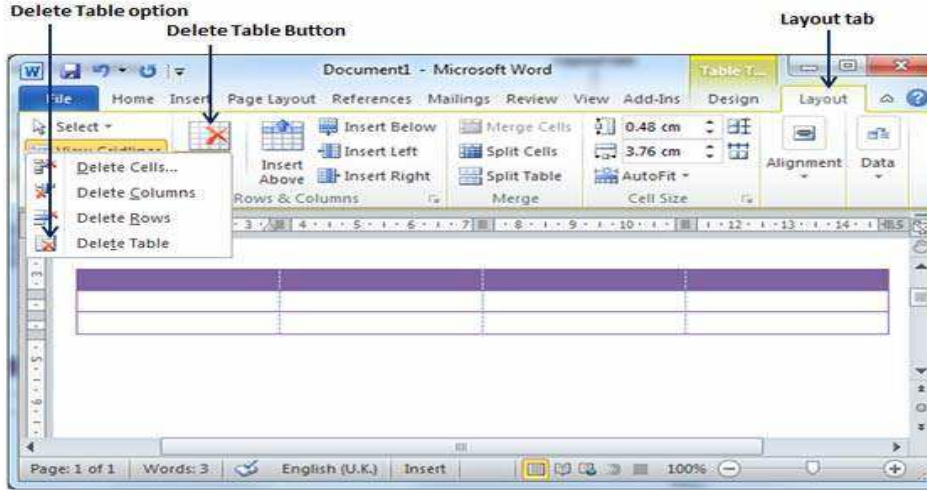
Following are the simple steps to delete columns from a table of a word document.

Step 1 – Click a column which you want to delete from the table and then click the **Layout tab**; it will show the following screen.



Step 2 – Click the **Layout tab**, and click the **Delete Column option** under the **Delete Table Button** to delete the selected column.

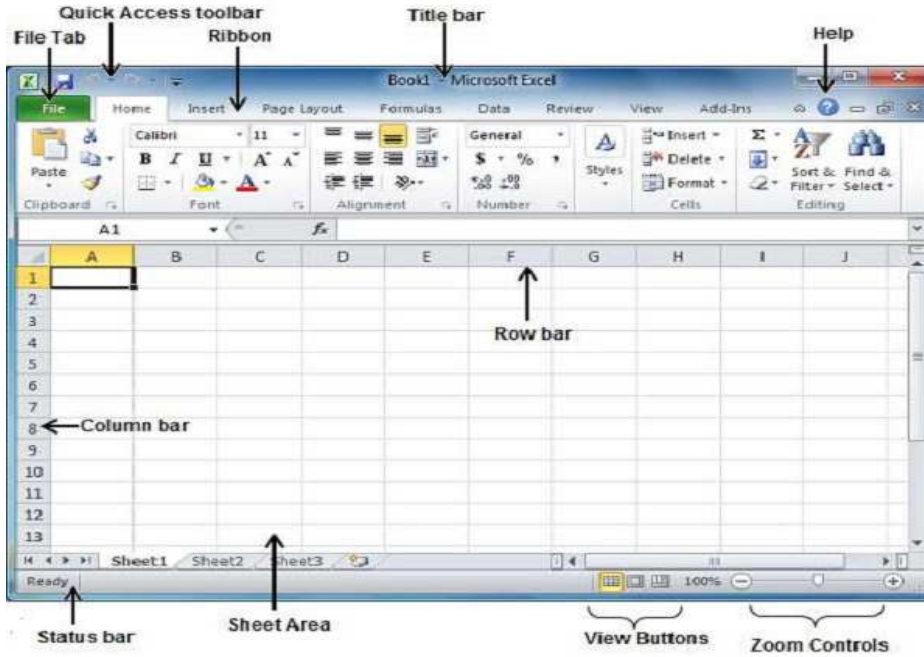

Delete Table Button
Layout tab
Document1 - Microsoft Word
Column1
Column2
Column3
Columns4


Delete Table option
Delete Table Button
Layout tab
Document1 - Microsoft Word
Delete Cells...
Delete Columns
Delete Rows
Delete Table

Chapter V

EXCEL

The following basic window appears when you start the excel application. Let us now understand the various important parts of this window.



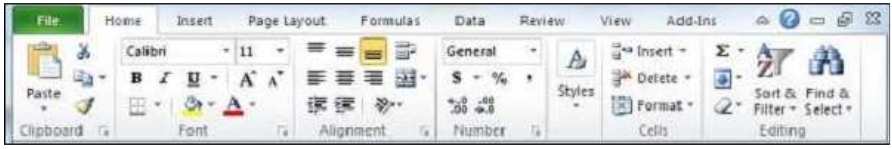
File Tab

The File tab replaces the Office button from Excel 2007. You can click it to check the **Backstage view,** where you come when you need to open or save files, create new sheets, print a sheet, and do other file-related operations.

Quick Access Toolbar

You will find this toolbar just above the **File tab** and its purpose is to provide a convenient resting place for the Excel's most frequently used commands. You can customize this toolbar based on your comfort.

Ribbon



Ribbon contains commands organized in three components –

- **Tabs** – They appear across the top of the Ribbon and contain groups of related commands. Home, Insert, Page Layout are the examples of ribbon tabs.
- **Groups** – They organize related commands; each group name appears below the group on the Ribbon. For example, group of commands related to fonts or group of commands related to alignment etc.
- **Commands** – Commands appear within each group as mentioned above.

Title Bar

This lies in the middle and at the top of the window. Title bar shows the program and the sheet titles.

Help

The **Help Icon** can be used to get excel related help anytime you like. This provides nice tutorial on various subjects related to excel.

Zoom Control

Zoom control lets you zoom in for a closer look at your text. The zoom control consists of a slider that you can slide left or right to zoom in or out. The + buttons can be clicked to increase or decrease the zoom factor.

View Buttons

The group of three buttons located to the left of the Zoom control, near the bottom of the screen, lets you switch among excel's various sheet views.

- **Normal Layout view** – This displays the page in normal view.
- **Page Layout view** – This displays pages exactly as they will appear when printed. This gives a full screen look of the document.
- **Page Break view** – This shows a preview of where pages will break when printed.

Sheet Area

The area where you enter data. The flashing vertical bar is called the **insertion point** and it represents the location where text will appear when you type.

Row Bar

Rows are numbered from 1 onwards and keeps on increasing as you keep entering data. Maximum limit is 1,048,576 rows.

Column Bar

Columns are numbered from A onwards and keeps on increasing as you keep entering data. After Z, it will start the series of AA, AB and so on. Maximum limit is 16,384 columns.

Status Bar

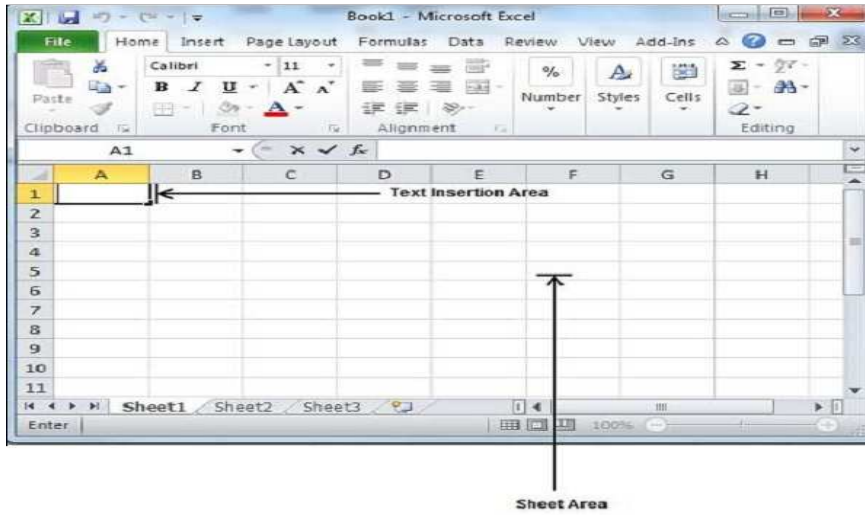
This displays the current status of the active cell in the worksheet. A cell can be in either of the fours states (a) Ready mode which indicates that the worksheet is ready to accept user inpu (b) Edit mode indicates that cell is editing mode, if it is not activated the you can activate editing mode by double-clicking on a cell (c) A cell enters into Enter mode when a user types data into a cell (d) Point mode triggers when a

formula is being entered using a cell reference by mouse pointing or the arrow keys on the keyboard.

Dialog Box Launcher

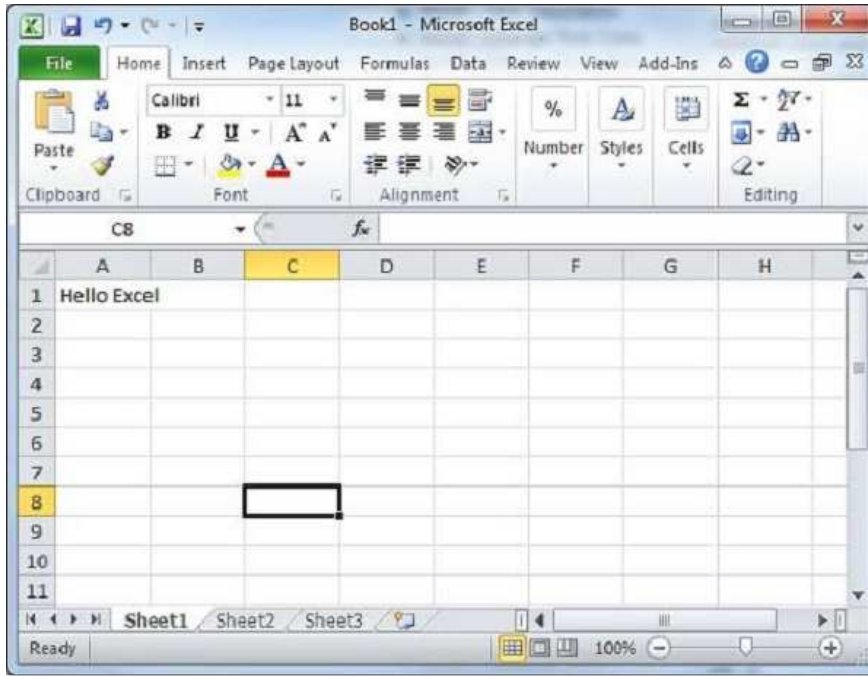
This appears as a very small arrow in the lower-right corner of many groups on the Ribbon. Clicking this button opens a dialog box or task pane that provides more options about the group.

Entering values in excel sheet is a child's play and this chapter shows how to enter values in an excel sheet. A new sheet is displayed by default when you open an excel sheet as shown in the below screen shot.



Sheet area is the place where you type your text. The flashing vertical bar is called the insertion point and it represents the location where text will appear when you type. When you click on a box then the box is highlighted. When you double click the box, the flashing vertical bar appears and you can start entering your data.

So, just keep your mouse cursor at the text insertion point and start typing whatever text you would like to type. We have typed only two words "Hello Excel" as shown below. The text appears to the left of the insertion point as you type.



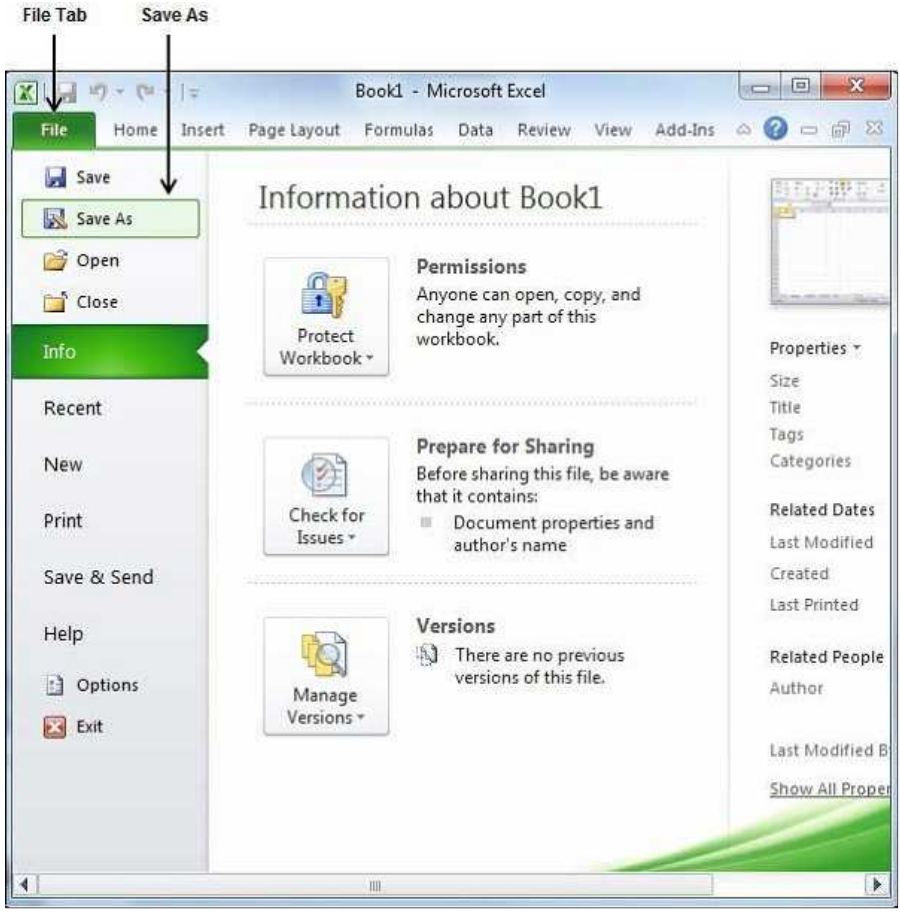
There are following three important points, which would help you while typing –

- Press Tab to go to next column.
- Press Enter to go to next row.
- Press Alt + Enter to enter a new line in the same column.

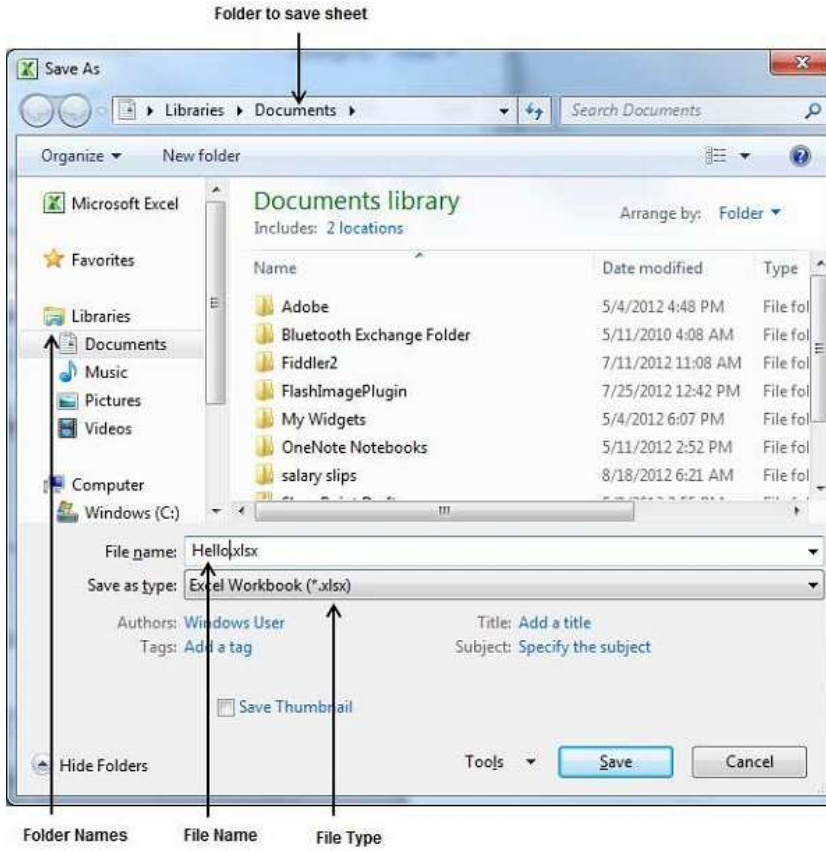
Saving New Sheet

Once you are done with typing in your new excel sheet, it is time to save your sheet/workbook to avoid losing work you have done on an Excel sheet. Following are the steps to save an edited excel sheet –

Step 1 – Click the **File tab** and select **Save As** option.



Step 2 – Select a folder where you would like to save the sheet, Enter file name, which you want to give to your sheet and Select a Save as type, by default it is **.xlsx** format.



Step 3 – Finally, click on **Save** button and your sheet will be saved with the entered name in the selected folder.

Saving New Changes

There may be a situation when you open an existing sheet and edit it partially or completely, or even you would like to save the changes in between editing of the sheet. If you want to save this sheet with the same name, then you can use either of the following simple options –

- Just press **Ctrl + S** keys to save the changes.

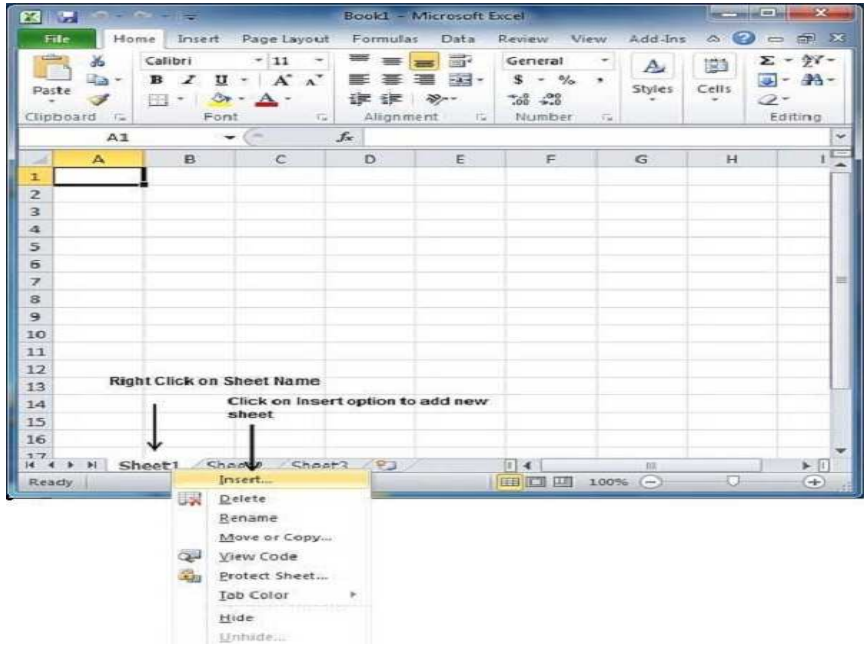
- Optionally, you can click on the floppy icon available at the top left corner and just above the **File tab**. This option will also save the changes.
- You can also use third method to save the changes, which is the **Save** option available just above the **Save As** option as shown in the above screen capture.

If your sheet is new and it was never saved so far, then with either of the three options, word would display you a dialogue box to let you select a folder, and enter sheet name as explained in case of saving new sheet.

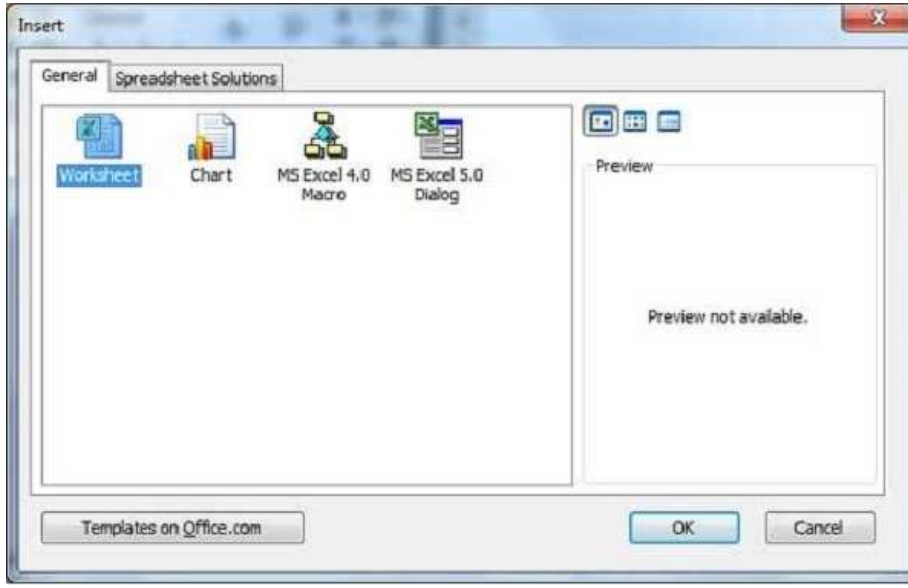
Creating New Worksheet

Three new blank sheets always open when you start Microsoft Excel. Below steps explain you how to create a new worksheet if you want to start another new worksheet while you are working on a worksheet, or you closed an already opened worksheet and want to start a new worksheet.

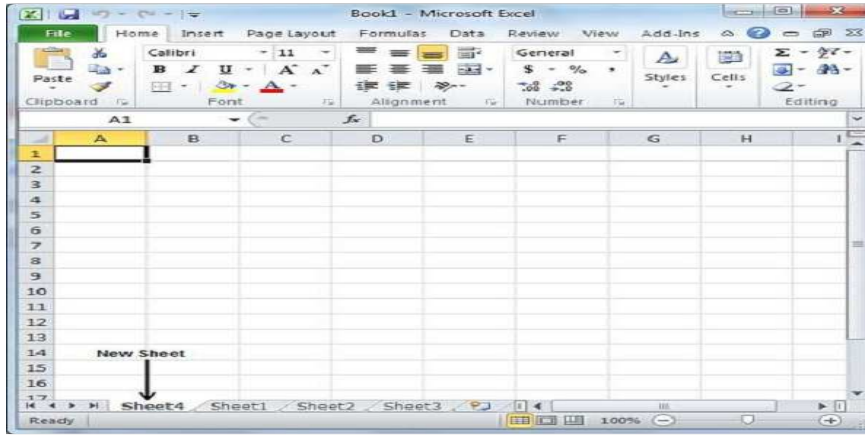
Step 1 – Right Click the Sheet Name and select Insert option.



Step 2 – Now you'll see the Insert dialog with select Worksheet option as selected from the general tab. Click the Ok button.



Now you should have your blank sheet as shown below ready to start typing your text.



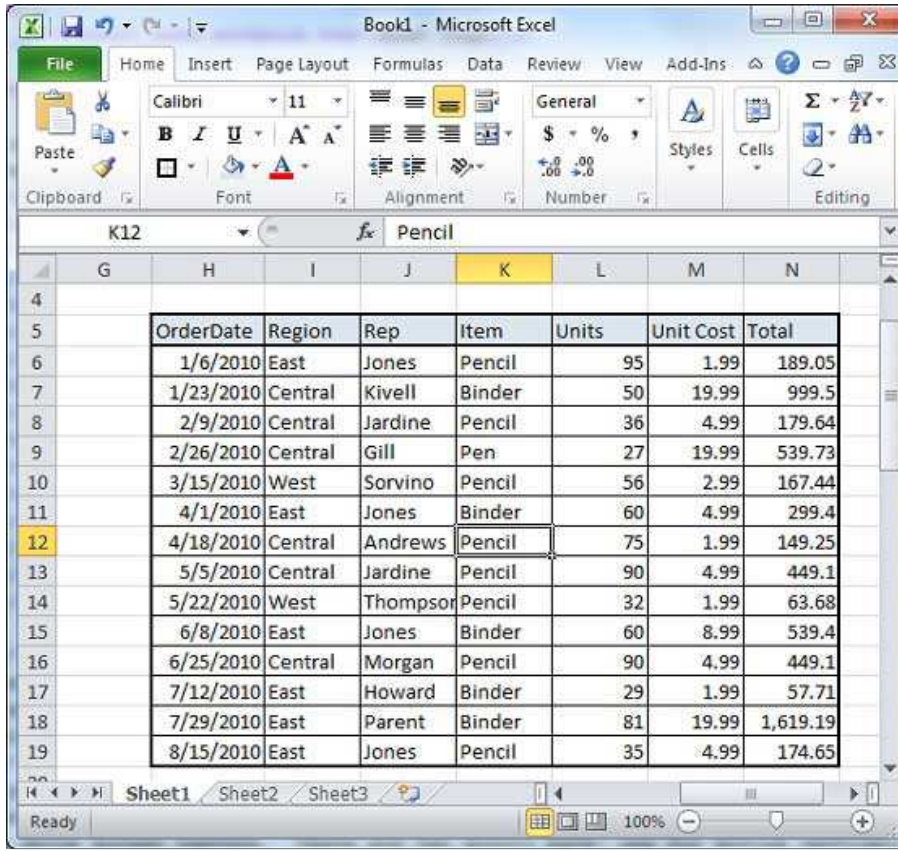
You can use a short cut to create a blank sheet anytime. Try using the **Shift+F11** keys and you will see a new blank sheet similar to the above sheet is opened.

Copy Worksheet

First of all, let us create some sample text before we proceed. Open a new excel sheet and type any data. We've shown a sample data in the screenshot.

OrderDate	Region	Rep	Item	Units	Unit Cost	Total
1/6/2010	East	Jones	Pencil	95	1.99	189.05
1/23/2010	Central	Kivell	Binder	50	19.99	999.5
2/9/2010	Central	Jardine	Pencil	36	4.99	179.64
2/26/2010	Central	Gill	Pen	27	19.99	539.73
3/15/2010	West	Sorvino	Pencil	56	2.99	167.44
4/1/2010	East	Jones	Binder	60	4.99	299.4

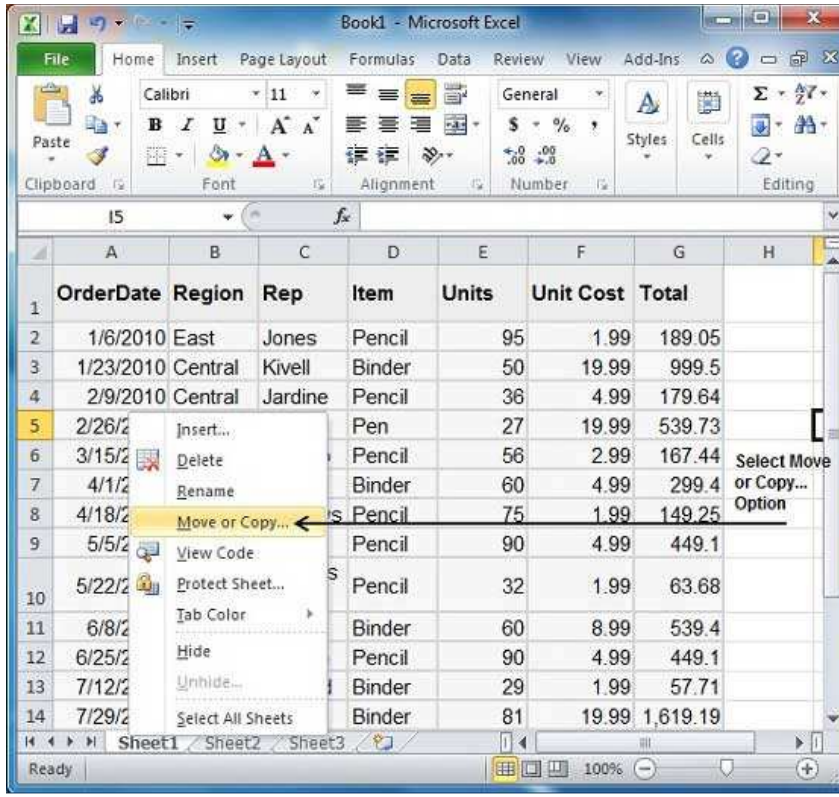
4/18/2010	Central	Andrews	Pencil	75	1.99	149.25
5/5/2010	Central	Jardine	Pencil	90	4.99	449.1
5/22/2010	West	Thompson	Pencil	32	1.99	63.68
6/8/2010	East	Jones	Binder	60	8.99	539.4
6/25/2010	Central	Morgan	Pencil	90	4.99	449.1
7/12/2010	East	Howard	Binder	29	1.99	57.71
7/29/2010	East	Parent	Binder	81	19.99	1,619.19
8/15/2010	East	Jones	Pencil	35	4.99	174.65



	G	H	I	J	K	L	M	N
4								
5		OrderDate	Region	Rep	Item	Units	Unit Cost	Total
6		1/6/2010	East	Jones	Pencil	95	1.99	189.05
7		1/23/2010	Central	Kivell	Binder	50	19.99	999.5
8		2/9/2010	Central	Jardine	Pencil	36	4.99	179.64
9		2/26/2010	Central	Gill	Pen	27	19.99	539.73
10		3/15/2010	West	Sorvino	Pencil	56	2.99	167.44
11		4/1/2010	East	Jones	Binder	60	4.99	299.4
12		4/18/2010	Central	Andrews	Pencil	75	1.99	149.25
13		5/5/2010	Central	Jardine	Pencil	90	4.99	449.1
14		5/22/2010	West	Thompsor	Pencil	32	1.99	63.68
15		6/8/2010	East	Jones	Binder	60	8.99	539.4
16		6/25/2010	Central	Morgan	Pencil	90	4.99	449.1
17		7/12/2010	East	Howard	Binder	29	1.99	57.71
18		7/29/2010	East	Parent	Binder	81	19.99	1,619.19
19		8/15/2010	East	Jones	Pencil	35	4.99	174.65

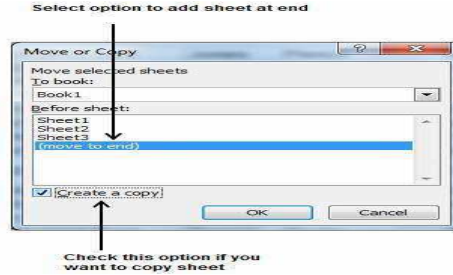
Here are the steps to copy an entire worksheet.

Step 1 – Right Click the **Sheet Name** and select the **Move or Copy** option.



	A	B	C	D	E	F	G	H
1	OrderDate	Region	Rep	Item	Units	Unit Cost	Total	
2	1/6/2010	East	Jones	Pencil	95	1.99	189.05	
3	1/23/2010	Central	Kivell	Binder	50	19.99	999.5	
4	2/9/2010	Central	Jardine	Pencil	36	4.99	179.64	
5	2/26/2			Pen	27	19.99	539.73	
6	3/15/2			Pencil	56	2.99	167.44	
7	4/1/2			Binder	60	4.99	299.4	
8	4/18/2			Pencil	75	1.99	149.25	
9	5/5/2			Pencil	90	4.99	449.1	
10	5/22/2			Pencil	32	1.99	63.68	
11	6/8/2			Binder	60	8.99	539.4	
12	6/25/2			Pencil	90	4.99	449.1	
13	7/12/2			Binder	29	1.99	57.71	
14	7/29/2			Binder	81	19.99	1,619.19	

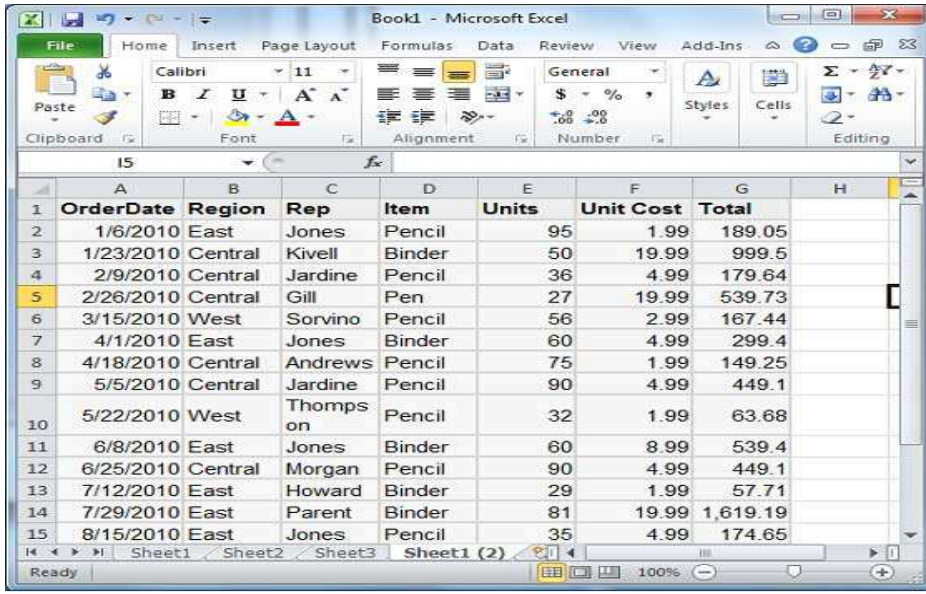
Step 2 – Now you'll see the Move or Copy dialog with select **Worksheet** option as selected from the general tab. Click the **Ok** button.



Select **Create a Copy** Checkbox to create a copy of the current sheet and Before sheet option as **(move to end)** so that new sheet gets created at the end.

Press the **Ok** Button.

Now you should have your copied sheet as shown below.



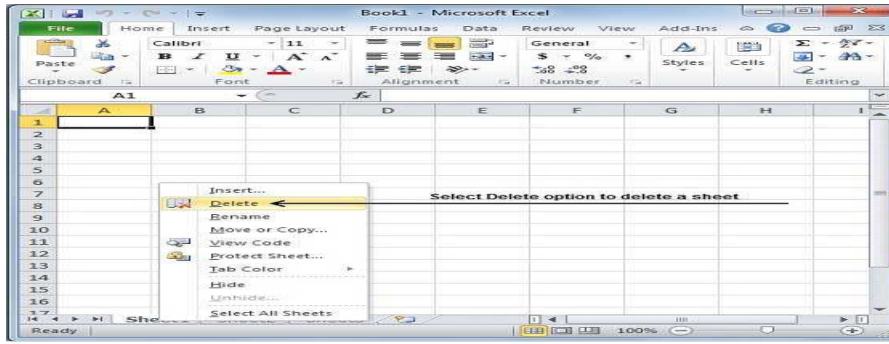
	A	B	C	D	E	F	G	H
1	OrderDate	Region	Rep	Item	Units	Unit Cost	Total	
2	1/6/2010	East	Jones	Pencil	95	1.99	189.05	
3	1/23/2010	Central	Kivell	Binder	50	19.99	999.5	
4	2/9/2010	Central	Jardine	Pencil	36	4.99	179.64	
5	2/26/2010	Central	Gill	Pen	27	19.99	539.73	
6	3/15/2010	West	Sorvino	Pencil	56	2.99	167.44	
7	4/1/2010	East	Jones	Binder	60	4.99	299.4	
8	4/18/2010	Central	Andrews	Pencil	75	1.99	149.25	
9	5/5/2010	Central	Jardine	Pencil	90	4.99	449.1	
10	5/22/2010	West	Thompson	Pencil	32	1.99	63.68	
11	6/8/2010	East	Jones	Binder	60	8.99	539.4	
12	6/25/2010	Central	Morgan	Pencil	90	4.99	449.1	
13	7/12/2010	East	Howard	Binder	29	1.99	57.71	
14	7/29/2010	East	Parent	Binder	81	19.99	1,619.19	
15	8/15/2010	East	Jones	Pencil	35	4.99	174.65	

You can rename the sheet by double clicking on it. On double click, the sheet name becomes editable. Enter any name say Sheet5 and press Tab or Enter Key.

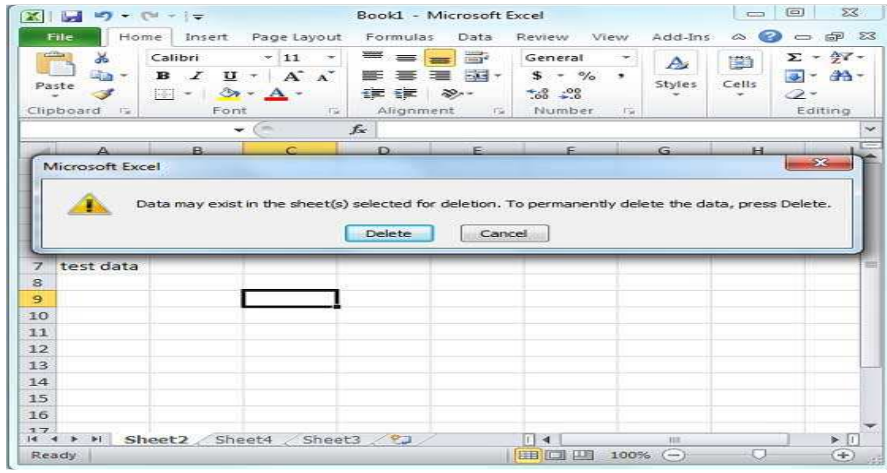
Delete Worksheet

Here is the step to delete a worksheet.

Step – Right Click the **Sheet Name** and select the **Delete** option.



Sheet will get deleted if it is empty, otherwise you'll see a confirmation message.



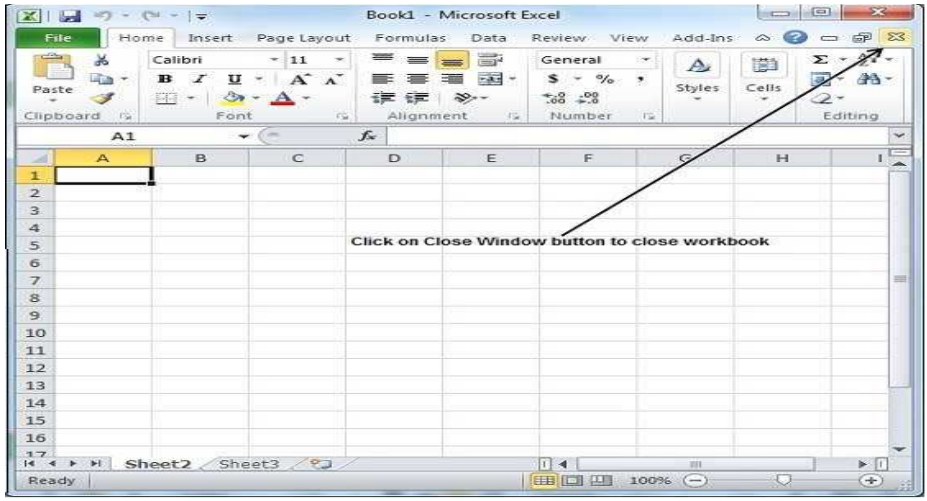
Press the **Delete** Button.

Now your worksheet will get deleted.

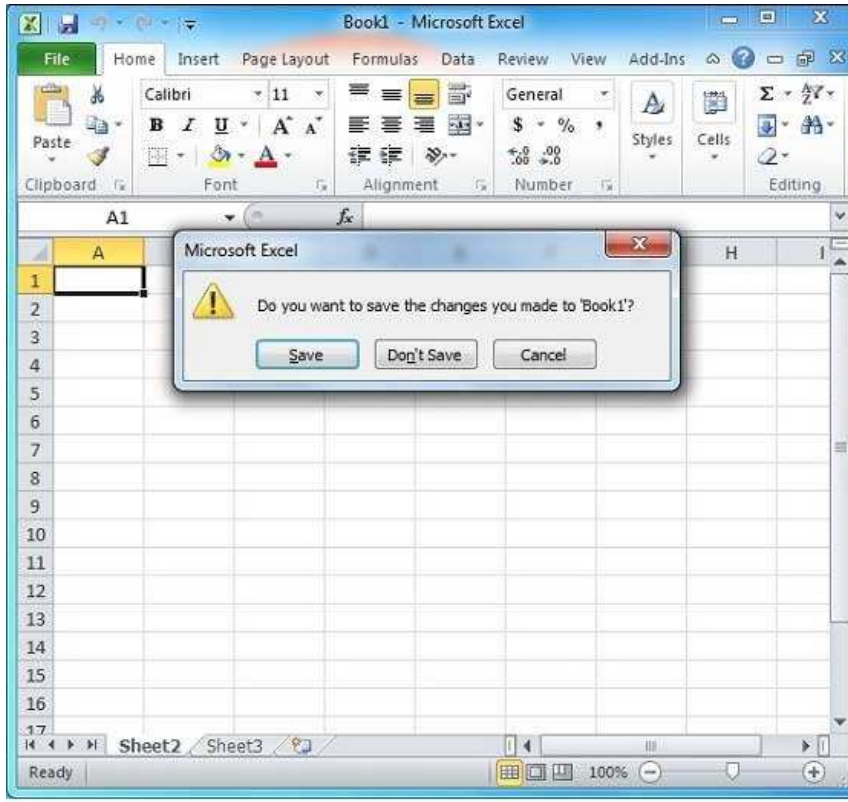
Close Workbook

Here are the steps to close a workbook.

Step 1 – Click the **Close Button** as shown below.



You'll see a confirmation message to save the workbook.



Step 2 – Press the **Save** Button to save the workbook as we did in MS Excel - Save Workbook chapter.

Now your worksheet will get closed.

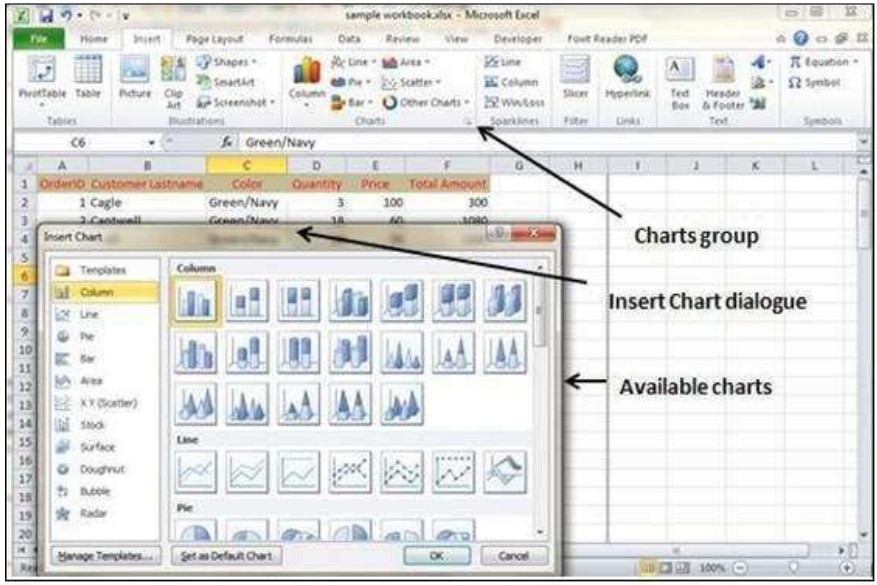
Charts

A chart is a visual representation of numeric values. Charts (also known as graphs) have been an integral part of spreadsheets. Charts generated by early spreadsheet products were quite crude, but thy have improved significantly over the years. Excel provides you with the tools to create a wide variety of highly customizable charts. Displaying data in a well-conceived chart can make your numbers more understandable. Because a chart presents a picture, charts are

particularly useful for summarizing a series of numbers and their interrelationships.

Types of Charts

There are various chart types available in MS Excel as shown in the below screen-shot.



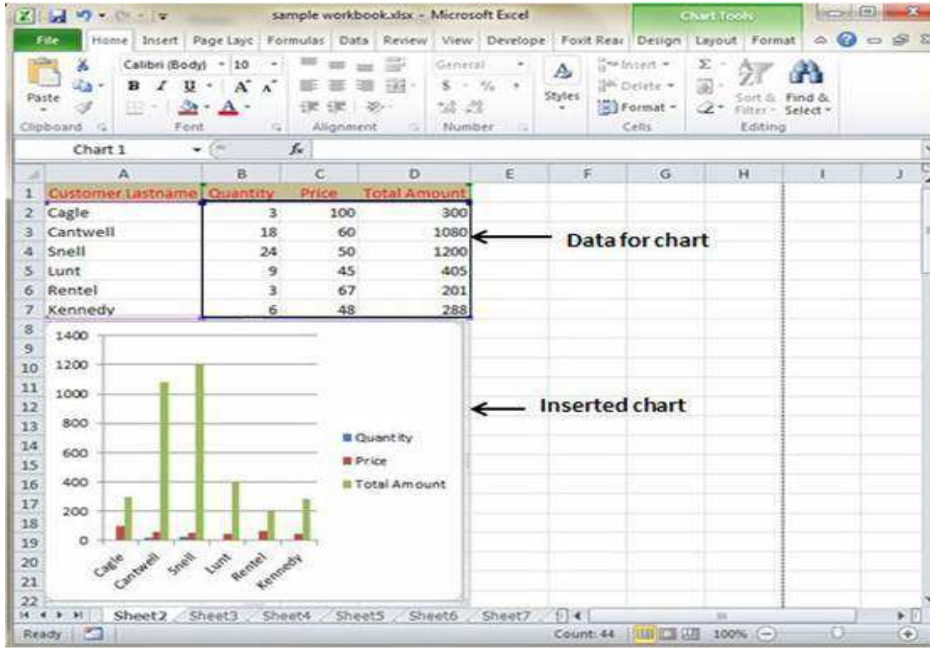
- **Column** – Column chart shows data changes over a period of time or illustrates comparisons among items.
- **Bar** – A bar chart illustrates comparisons among individual items.
- **Pie** – A pie chart shows the size of items that make up a data series, proportional to the sum of the items. It always shows only one data series and is useful when you want to emphasize a significant element in the data.
- **Line** – A line chart shows trends in data at equal intervals.

- **Area** – An area chart emphasizes the magnitude of change over time.
- **X Y Scatter** – An xy (scatter) chart shows the relationships among the numeric values in several data series, or plots two groups of numbers as one series of xy coordinates.
- **Stock** – This chart type is most often used for stock price data, but can also be used for scientific data (for example, to indicate temperature changes).
- **Surface** – A surface chart is useful when you want to find the optimum combinations between two sets of data. As in a topographic map, colors and patterns indicate areas that are in the same range of values.
- **Doughnut** – Like a pie chart, a doughnut chart shows the relationship of parts to a whole; however, it can contain more than one data series.
- **Bubble** – Data that is arranged in columns on a worksheet, so that x values are listed in the first column and corresponding y values and bubble size values are listed in adjacent columns, can be plotted in a bubble chart.
- **Radar** – A radar chart compares the aggregate values of a number of data series.

Creating Chart

To create charts for the data by below mentioned steps.

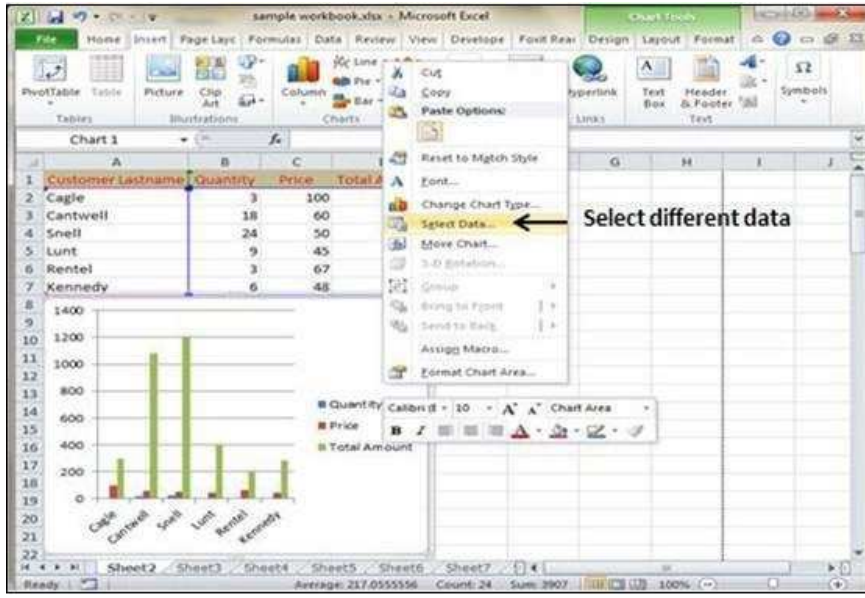
- Select the data for which you want to create the chart.
- Choose **Insert Tab » Select the chart or click on the Chart group** to see various chart types.
- Select the chart of your choice and click OK to generate the chart.



Editing Chart

You can edit the chart at any time after you have created it.

- You can select the different data for chart input with **Right click on chart » Select data**. Selecting new data will generate the chart as per the new data, as shown in the below screen-shot.



- You can change the X axis of the chart by giving different inputs to X-axis of chart.
- You can change the Y axis of chart by giving different inputs to Y-axis of chart.

MS Excel Keyboard Short-cuts

MS Excel offers many keyboard short-cuts. If you are familiar with windows operating system, you should be aware of most of them. Below is the list of all the major shortcut keys in Microsoft Excel.

- **Ctrl + A** – Selects all contents of the worksheet.
- **Ctrl + B** – Bold highlighted selection.
- **Ctrl + I** – Italicizes the highlighted selection.
- **Ctrl + K** – Inserts link.
- **Ctrl + U** – Underlines the highlighted selection.

- **Ctrl + 1** – Changes the format of selected cells.
- **Ctrl + 5** – Strikethrough the highlighted selection.
- **Ctrl + P** – Brings up the print dialog box to begin printing.
- **Ctrl + Z** – Undo last action.
- **Ctrl + F3** – Opens Excel Name Manager.
- **Ctrl + F9** – Minimizes the current window.
- **Ctrl + F10** – Maximize currently selected window.
- **Ctrl + F6** – Switches between open workbooks or windows.
- **Ctrl + Page up** – Moves between Excel work sheets in the same Excel document.
- **Ctrl + Page down** – Moves between Excel work sheets in the same Excel document.
- **Ctrl + Tab** – Moves between Two or more open Excel files.
- **Alt + =** – Creates a formula to sum all of the above cells
- **Ctrl + '** – Inserts the value of the above cell into cell currently selected.
- **Ctrl + Shift + !** – Formats the number in comma format.
- **Ctrl + Shift + $** – Formats the number in currency format.
- **Ctrl + Shift + #** – Formats the number in date format.
- **Ctrl + Shift + %** – Formats the number in percentage format.
- **Ctrl + Shift + ^** – Formats the number in scientific format.
- **Ctrl + Shift + @** – Formats the number in time format.
- **Ctrl + Arrow key** – Moves to the next section of text.
- **Ctrl + Space** – Selects the entire column.

- **Shift + Space** – Selects the entire row.
- **Ctrl + -** – Deletes the selected column or row.
- **Ctrl + Shift + =** – Inserts a new column or row.
- **Ctrl + Home** – Moves to cell A1.
- **Ctrl + ~** – Switches between showing Excel formulas or their values in cells.
- **F2** – Edits the selected cell.
- **F3** – After a name has been created F3 will paste names.
- **F4** – Repeat last action. For example, if you changed the color of text in another cell pressing F4 will change the text in cell to the same color.
- **F5** – Goes to a specific cell. For example, C6.
- **F7** – Spell checks the selected text or document.
- **F11** – Creates chart from the selected data.
- **Ctrl + Shift + ;** – Enters the current time.
- **Ctrl + ;** – Enters the current date.
- **Alt + Shift + F1** – Inserts New Worksheet.
- **Alt + Enter** – While typing text in a cell pressing Alt + Enter will move to the next line allowing for multiple lines of text in one cell.
- **Shift + F3** – Opens the Excel formula window.
- **Shift + F5** – Brings up the search box.

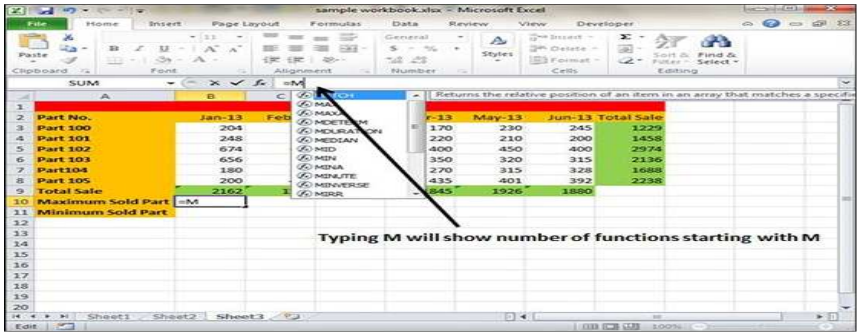
Functions in Formula

Many formulas you create use available worksheet functions. These functions enable you to greatly enhance the power of your formulas and perform calculations that are difficult if you use only the operators.

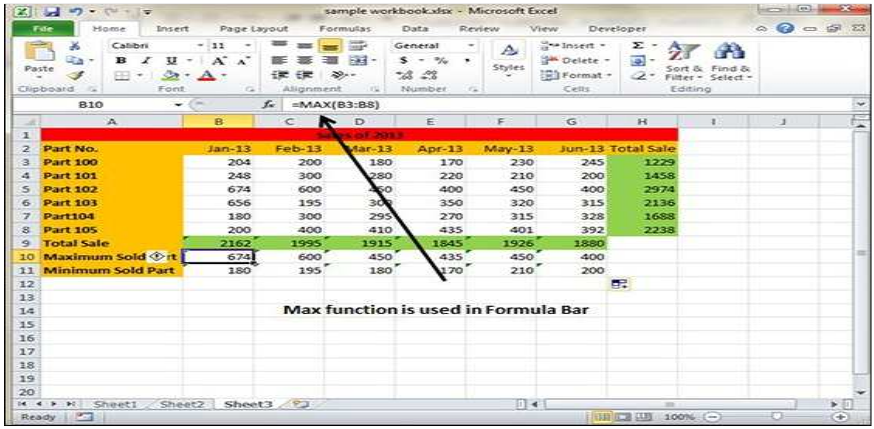
For example, you can use the LOG or SIN function to calculate the Logarithm or Sin ratio. You can't do this complicated calculation by using the mathematical operators alone.

Using Functions

When you type = sign and then type any alphabet you will see the searched functions as below.

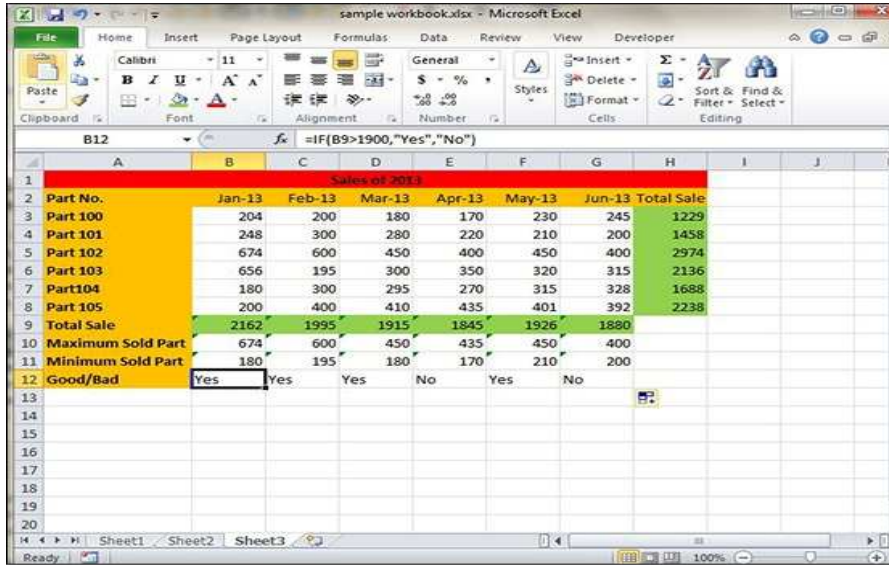


Suppose you need to determine the largest value in a range. A formula can't tell you the answer without using a function. We will use formula that uses the MAX function to return the largest value in the range B3:B8 as **=MAX(A1:D100)**.



Another example of functions. Suppose you want to find if the cell of month is greater than 1900 then we can give Bonus to Sales

representative. The we can achieve it with writing formula with IF functions as **=IF(B9>1900,"Yes","No")**



	A	B	C	D	E	F	G	H
1	Sales of 2013							
2	Part No.	Jan-13	Feb-13	Mar-13	Apr-13	May-13	Jun-13	Total Sale
3	Part 100	204	200	180	170	230	245	1229
4	Part 101	248	300	280	220	210	200	1458
5	Part 102	674	600	450	400	450	400	2974
6	Part 103	656	195	300	350	320	315	2136
7	Part104	180	300	295	270	315	328	1688
8	Part 105	200	400	410	435	401	392	2238
9	Total Sale	2162	1995	1915	1845	1926	1880	
10	Maximum Sold Part	674	600	450	435	450	400	
11	Minimum Sold Part	180	195	180	170	210	200	
12	Good/Bad	Yes	Yes	Yes	No	Yes	No	

Function Arguments

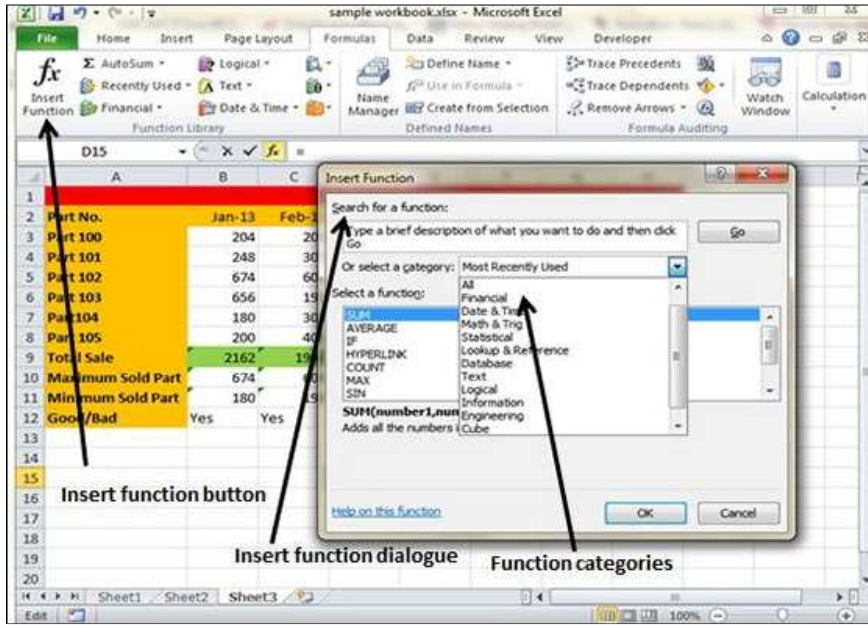
In the above examples, you may have noticed that all the functions used parentheses. The information inside the parentheses is the list of arguments.

Functions vary in how they use arguments. Depending on what it has to do, a function may use.

- **No arguments** – Examples – Now(), Date(), etc.
- **One argument** – UPPER(), LOWER(), etc.
- **A fixed number of arguments** – IF(), MAX(), MIN(), AVERGAGE(), etc.
- **Infinite number of arguments**
- **Optional arguments**

Built In Functions

MS Excel has many built in functions, which we can use in our formula. To see all the functions by category, choose Formulas Tab » Insert Function. Then Insert function Dialog appears from which we can choose the function.



Functions by Categories

Let us see some of the built in functions in MS Excel.

- **Text Functions**
    - **LOWER** – Converts all characters in a supplied text string to lower case
    - **UPPER** – Converts all characters in a supplied text string to upper case
    - **TRIM** – Removes duplicate spaces, and spaces at the start and end of a text string

- **CONCATENATE** – Joins together two or more text strings.
- **LEFT** – Returns a specified number of characters from the start of a supplied text string.
- **MID** – Returns a specified number of characters from the middle of a supplied text string
- **RIGHT** – Returns a specified number of characters from the end of a supplied text string.
- **LEN** – Returns the length of a supplied text string
- **FIND** – Returns the position of a supplied character or text string from within a supplied text string (case-sensitive).

- **Date & Time**
  - **DATE** – Returns a date, from a user-supplied year, month and day.
  - **TIME** – Returns a time, from a user-supplied hour, minute and second.
  - **DATEVALUE** – Converts a text string showing a date, to an integer that represents the date in Excel's date-time code.
  - **TIMEVALUE** – Converts a text string showing a time, to a decimal that represents the time in Excel.
  - **NOW** – Returns the current date & time.
  - **TODAY** – Returns today's date.
- **Statistical**
  - **MAX** – Returns the largest value from a list of supplied numbers.

- **MIN** – Returns the smallest value from a list of supplied numbers.
- **AVERAGE** – Returns the Average of a list of supplied numbers.
- **COUNT** – Returns the number of numerical values in a supplied set of cells or values.
- **COUNTIF** – Returns the number of cells (of a supplied range), that satisfies a given criteria.
- **SUM** – Returns the sum of a supplied list of numbers

- **Logical**
  - **AND** – Tests a number of user-defined conditions and returns TRUE if ALL of the conditions evaluate to TRUE, or FALSE otherwise
  - **OR** – Tests a number of user-defined conditions and returns TRUE if ANY of the conditions evaluate to TRUE, or FALSE otherwise.
  - **NOT** – Returns a logical value that is the opposite of a user supplied logical value or expression i.e. returns FALSE if the supplied argument is TRUE and returns TRUE if the supplied argument is FAL

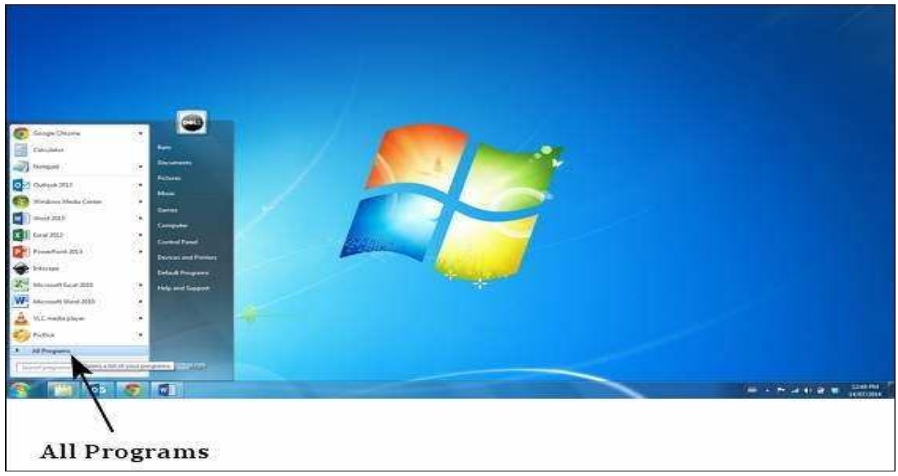
- **Math & Trig**
  - **ABS** – Returns the absolute value (i.e. the modulus) of a supplied number.
  - **SIGN** – Returns the sign (+1, -1 or 0) of a supplied number.
  - **SQRT** – Returns the positive square root of a given number.
  - **MOD** – Returns the remainder from a division between two supplied numbers.

PowerPoint

We will understand how to start PowerPoint 2010 application in simple steps. To access PowerPoint 2010, you must have Microsoft Office 2010 installed in your PC. Only Office 2010 Home and Student, Home and Business, Standard, Professional and Professional Plus packages have PowerPoint included in them. Other packages may have a viewer, but you cannot create presentations with them.

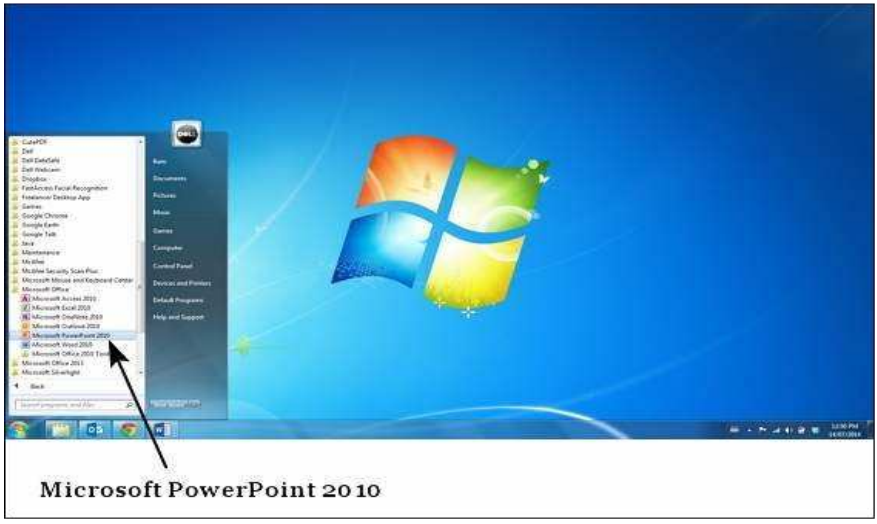
Step 1 – Click the **Start** button.

Step 2 – Click **All Programs** option from the menu.

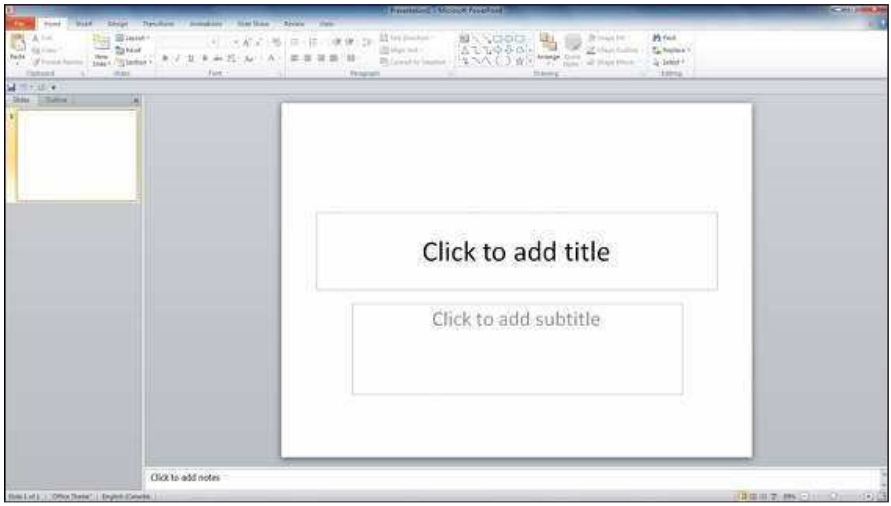


Step 3 – Search for **Microsoft Office** from the sub menu and click it.

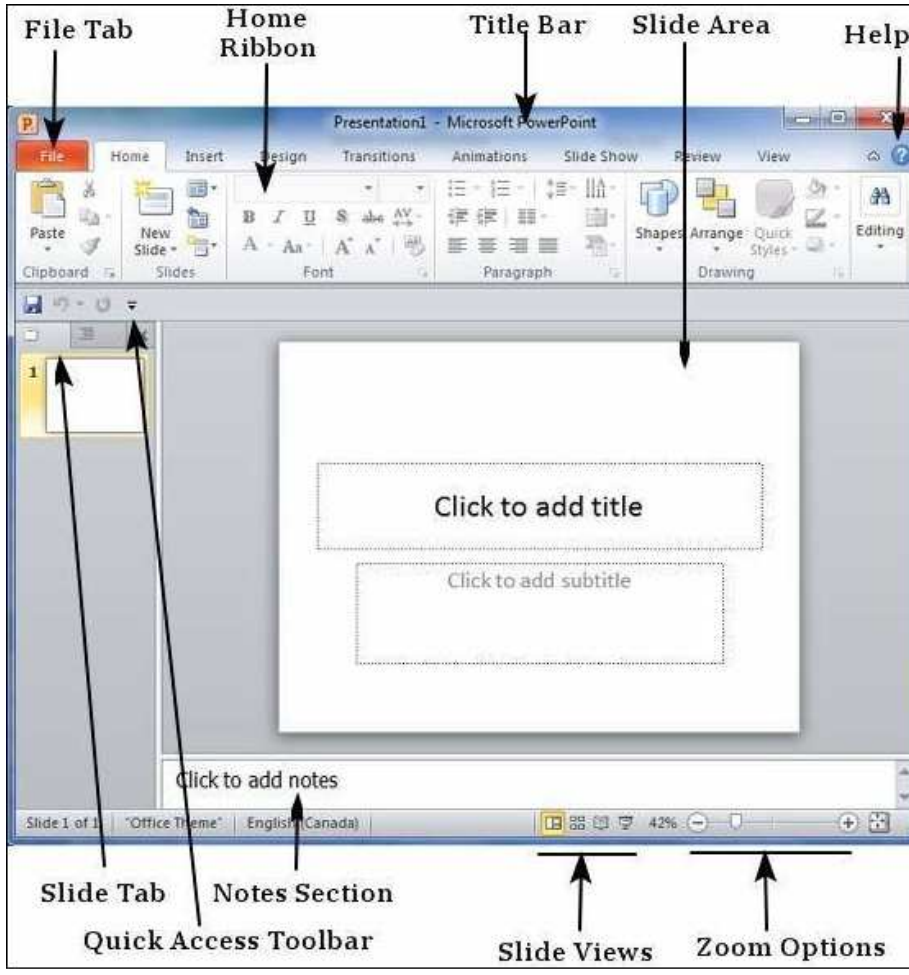
Step 4 – Search for **Microsoft PowerPoint 2010** from the submenu and click it.



This will launch the Microsoft PowerPoint 2010 application and you will see the following presentation window.



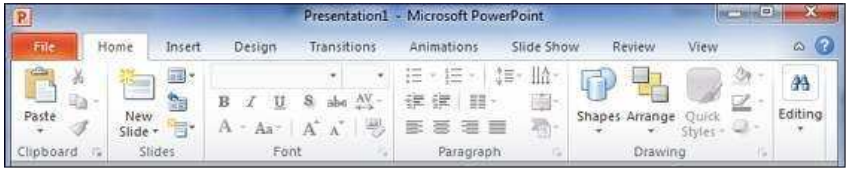
The following screenshot shows the various areas in a standard PowerPoint file. It is important to familiarize yourself with these areas as it makes learning and using PowerPoint easier.



File Tab

This tab opens the **Backstage** view which basically allows you to manage the file and settings in PowerPoint. You can save presentations, open existing ones and create new presentations based on blank or predefined templates. The other file related operations can also be executed from this view.

Ribbon



The ribbon contains three components –

- **Tabs** – They appear across the top of the Ribbon and contain groups of related commands. **Home**, **Insert**, **Page Layout** are examples of ribbon tabs.
- **Groups** – They organize related commands; each group name appears below the group on the Ribbon. For example, a group of commands related to fonts or a group of commands related to alignment, etc.
- **Commands** – Commands appear within each group as mentioned above.

Title Bar

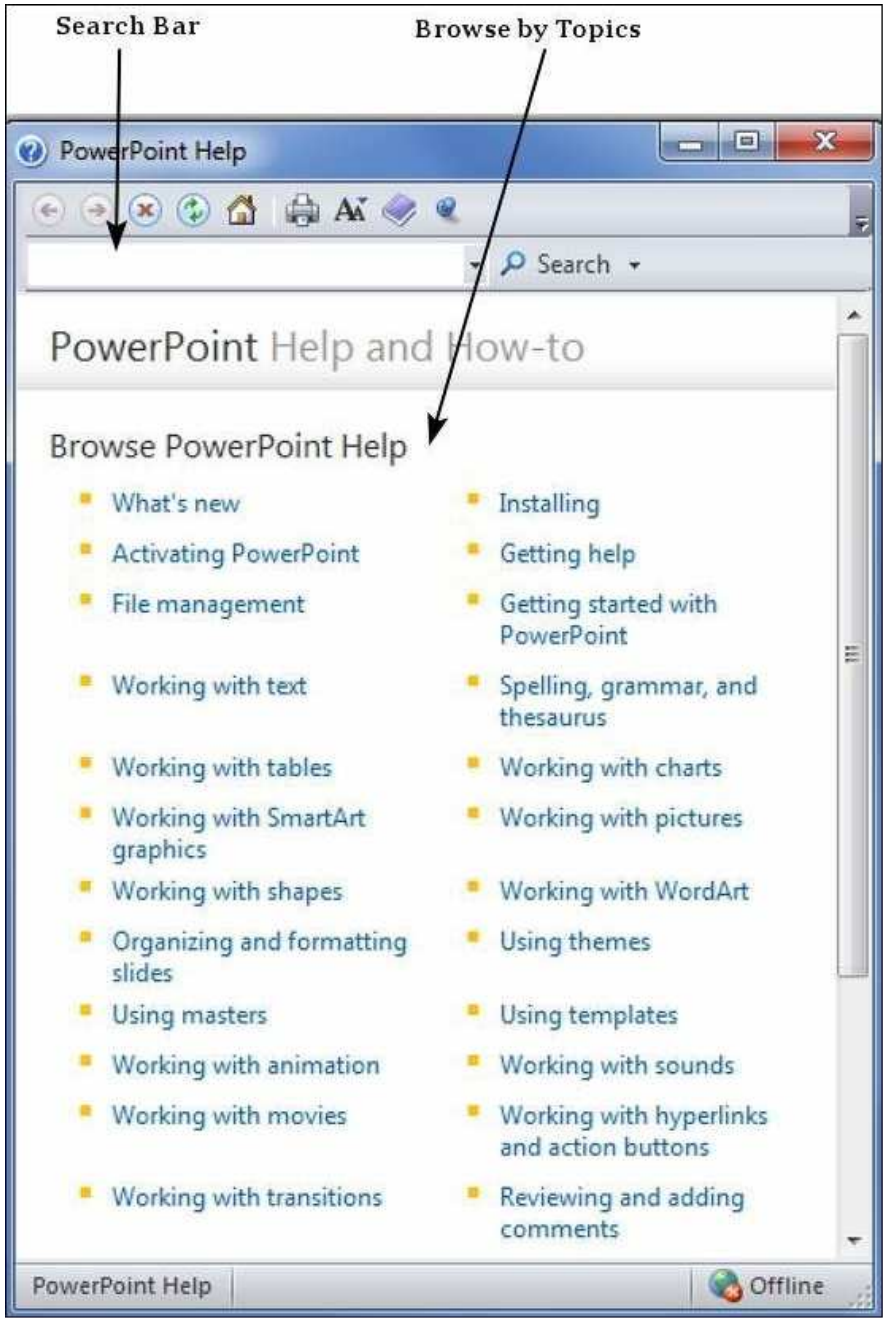
This is the top section of the window. It shows the name of the file followed by the name of the program which in this case is Microsoft PowerPoint.

Slide Area

This is the area where the actual slide is created and edited. You can add, edit and delete text, images, shapes and multimedia in this section.

Help

The Help Icon can be used to get PowerPoint related help anytime you need. Clicking on the "?" opens the PowerPoint Help window where you have a list of common topics to browse from. You can also search for specific topics from the search bar at the top.


Search Bar
Browse by Topics
PowerPoint Help
Search
PowerPoint Help and How-to
Browse PowerPoint Help
What's new
Activating PowerPoint
File management
Working with text
Working with tables
Working with SmartArt graphics
Working with shapes
Organizing and formatting slides
Using masters
Working with animation
Working with movies
Working with transitions
Installing
Getting help
Getting started with PowerPoint
Spelling, grammar, and thesaurus
Working with charts
Working with pictures
Working with WordArt
Using themes
Using templates
Working with sounds
Working with hyperlinks and action buttons
Reviewing and adding comments
PowerPoint Help
Offline

S.No	Command & Description
1	**Save** This allows you to save a new file or an existing file in standard format. If you are working on a previously saved file this will save the new changes in the same file format. If you are working on a new file, this command would be similar to the Save As command.
2	**Save As** Allows you to specify the file name and the file type before saving the file.
3	**Open** Allows you to open new PowerPoint files.
4	**Close** Allows you to close an existing file.
5	**Info** Displays the information about the current file.
6	**Recent** Lists series of recently viewed or edited PowerPoint files.
7	**New** Allows you to create a new file using blank or pre-defined templates.

8	**Print** Allows you to select the printer settings and print the presentation.
9	**Save & Send** Allows you to share your presentation with larger audience via emails, web, cloud services, etc.
10	**Help** Provides access to PowerPoint Help.
11	**Options** Allows you to set various options related to PowerPoint program.
12	**Exit** Closes the presentation and exits the program.

Zoom Options

The zoom control lets you zoom in for a closer look at your text. The zoom control consists of a slider that you can slide left or right to zoom in or out, you can click on the - and + buttons to increase or decrease the zoom factor. The maximum zoom supported by PowerPoint is 400% and the 100% is indicated by the mark in the middle.

Slide Views

The group of four buttons located to the left of the Zoom control, near the bottom of the screen, lets you switch between PowerPoint views.

- **Normal Layout view** – This displays page in normal view with the slide on the right and a list of thumbnails to the left. This

view allows you to edit individual slides and also rearrange them.

- **Slide Sorter view** – This displays all the slides as a matrix. This view only allows you to rearrange the slides but not edit the contents of each slide.
- **Reading View** – This view is like a slideshow with access to the Windows task bar in case you need to switch windows. However, like the slideshow you cannot edit anything in this view.

Notes Section

This sections allows you to add notes for the presentation. These notes will not be displayed on the screen during the presentation; these are just quick reference for the presenter.

Quick Access Toolbar

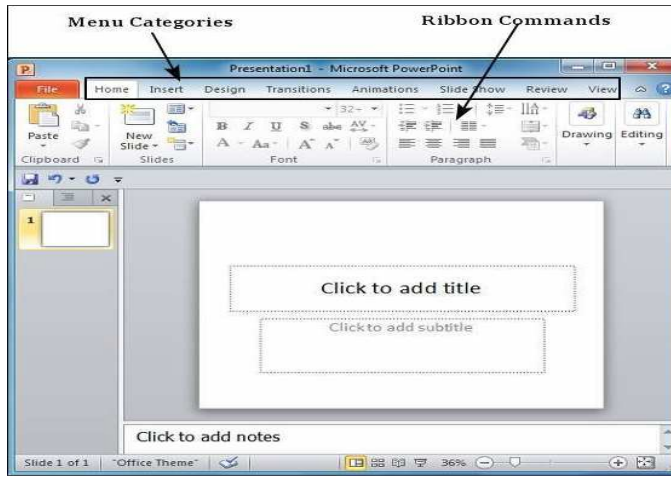
The Quick Access Toolbar is located just under the ribbon. This toolbar offers a convenient place to group the most commonly used commands in PowerPoint. You can customize this toolbar to suit your needs.

Slide Tab

This section is available only in the Normal view. It displays all the slides in sequence. You can add, delete and reorder slides from this section.
Various commands under the first pane are described in the table below –

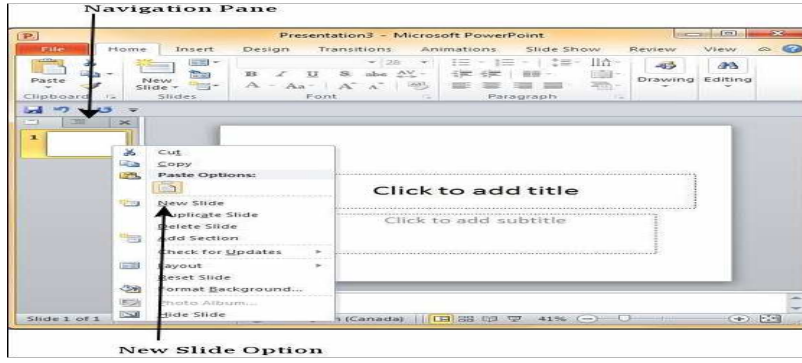
- **Second Pane** – This is the subcommands pane. This will list all the commands related to the main command you choose in the first pane. For example, if you select Print in the first pane, you get to choose the printer and adjust the print settings in the second pane.
- **Third Pane** – This is the preview or file information page. Depending on the command and the subcommand you select, this pane will either display the properties of the file or give you a preview of the file.

- PowerPoint offers a host of tools that will aid you in creating a presentation. These tools are organized logically into various ribbons in PowerPoint. The table below describes the various commands you can access from the different menus.

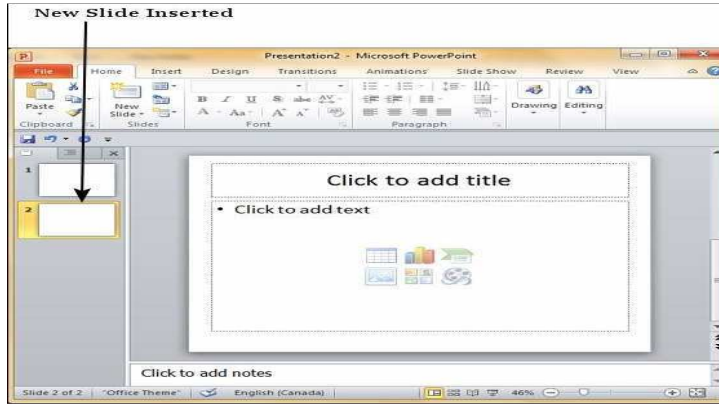


Menu Category	Ribbon Commands
Home	Clipboard functions, manipulating slides, fonts, paragraph settings, drawing objects and editing functions.
Insert	Insert tables, pictures, images, shapes, charts, special texts, multimedia and symbols.
Design	Slide setup, slide orientation, presentation themes and background.

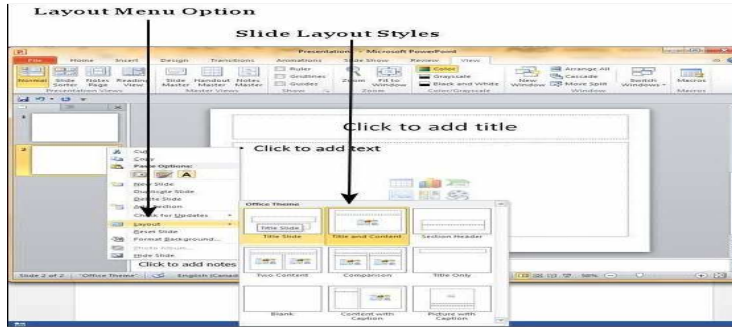
Transitions	Commands related to slide transitions.
Animations	Commands related to animation within the individual slides.
Slide Show	Commands related to slideshow set up and previews.
Review	Proofing content, language selection, comments and comparing presentations.
View	Commands related to presentation views, Master slides, color settings and window arrangements.

- Besides these depending on the objects selected in the slide, there are other menu tabs that get enabled.
- we will understand how to add new slides in an existing presentation. Here are the steps that allow you to insert a new slide in the deck –
- **Step 1** – Right-click in the **Navigation Pane** under any existing slide and click on the **New Slide** option.
- 

- **Step 2** – The new slide is inserted. You can now change the layout of this slide to suit your design requirements.



- **Step 3** – To change the slide layout, right-click on the newly inserted slide and go to the **Layout** option where you can choose from the existing layout styles available to you.

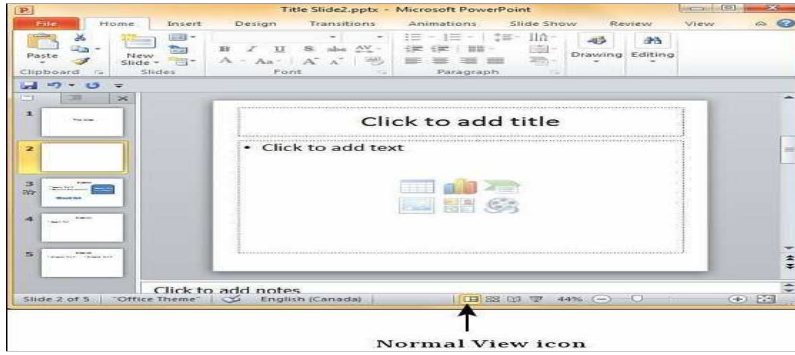


- You can follow the same steps to insert a new slide in between existing slides or at the end on the slide list.
- When we insert a new slide, it inherits the layout of its previous slide with one exception. If you are inserting a new slide after the first slide (**Title** slide), the subsequent slide will have the **Title and Content** layout.
- You will also notice that if you right-click in the first step without selecting any slide the menu options you get are different, although you can insert a new slide from this menu too.

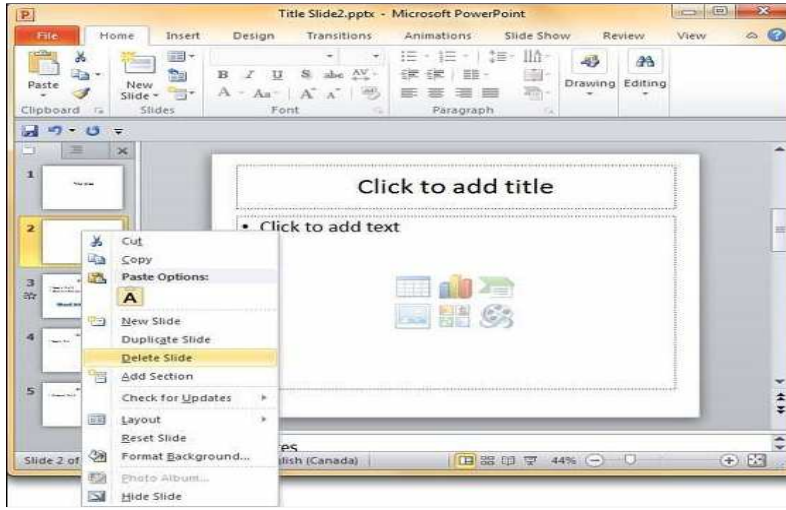
- There are times while building a slide deck, you may need to delete some slides. This can be done easily from PowerPoint. You can delete the slides from the Normal view as well as the Slide Sorter view. In each view, you can delete the slides in two ways.

- Deleting from Normal View

- **Step 1** – Go to the Normal view.

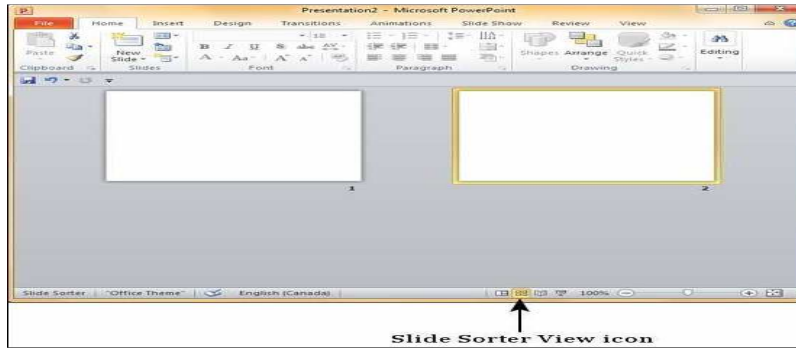


- **Step 2** – Right-click on the slide to be deleted and select the **Delete Slide** option.

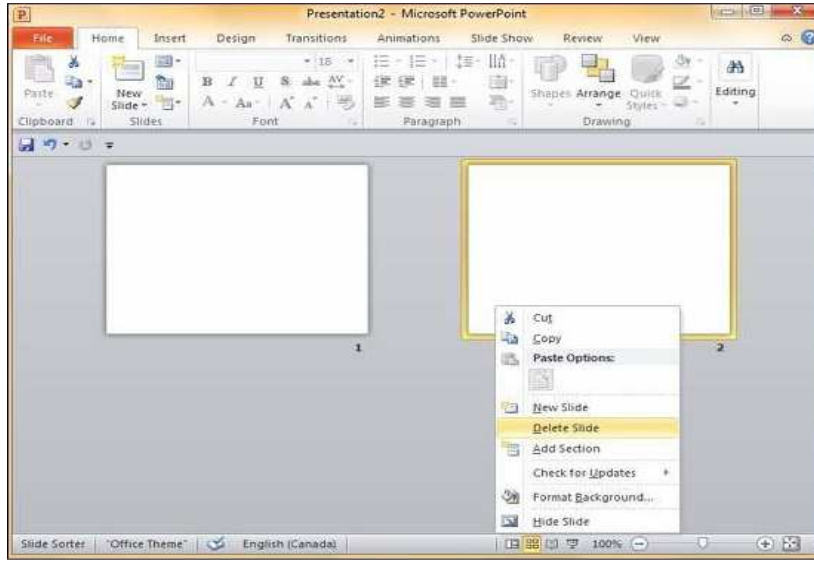


- Alternately, you can select the slide and press the **Delete** button on your key board.

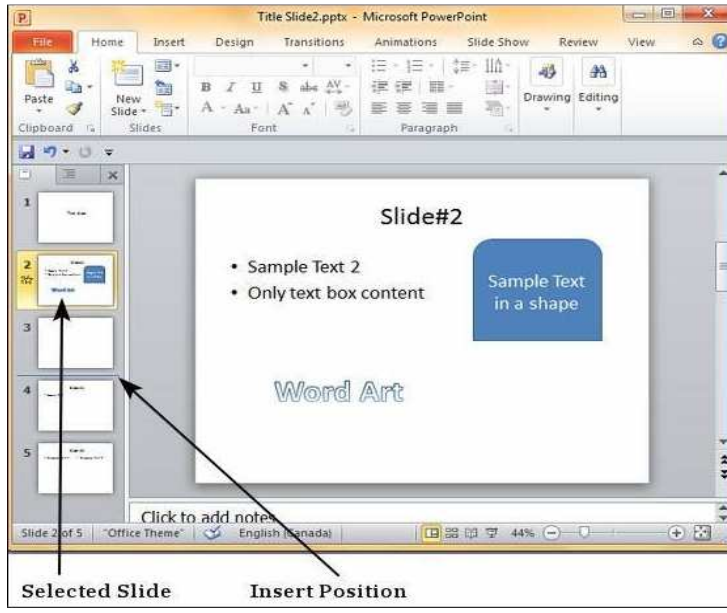
- Deleting from Slide Sorter View
- Let us now understand how to deleted slides from the Slide Sorter View.
- **Step 1** – Go to the Slide Sorter view.



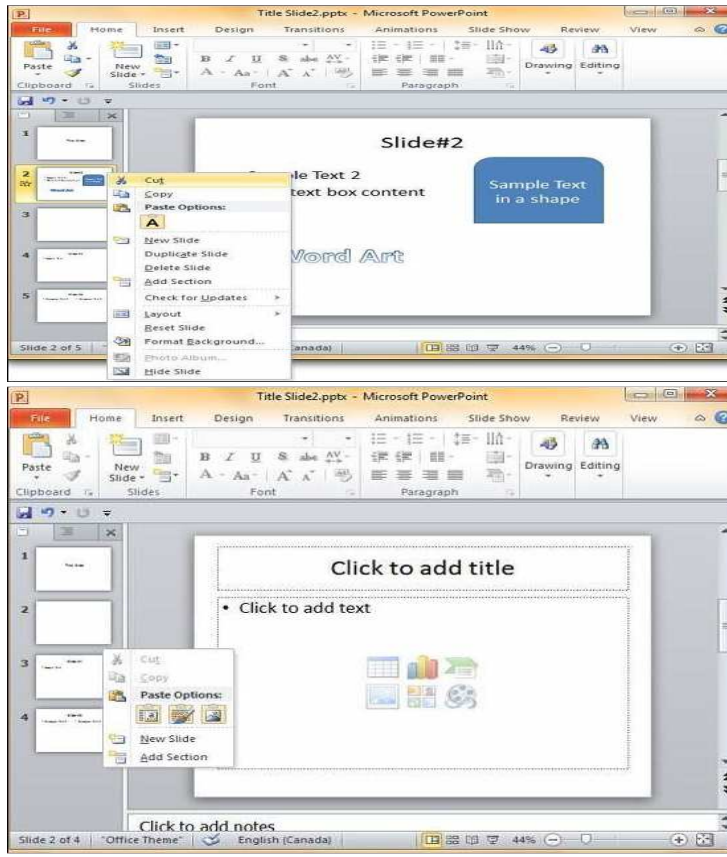
- **Step 2** – Right-click on the slide to be deleted and select the **Delete Slide** option.



-

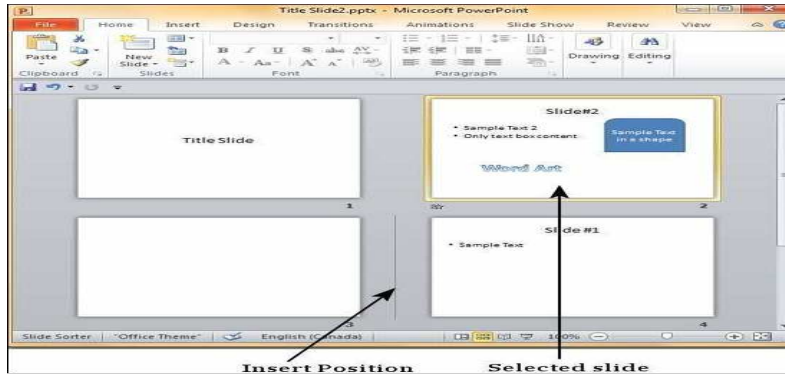
- Alternately, you can select the slide and press the **Delete** button on your key board.
- Rearranging slides is important when it comes to organizing the overall presentation flow. While it is vital that you get the right content in every slide, it is equally important that you are able to present them in a format that makes it easier for the audience to understand the content too; most times this will require rearranging the slides.
- You can rearrange slides from two views in PowerPoint - **Normal View** and **Slide Sorter View**. Given below are the steps to rearrange slides from different views.
- Normal View
- **Step 1** – Select the slide to be moved.
- **Step 2** – Left click on the slide and drag it to the position in the sequence where you want to place it. PowerPoint will indicate the insert position with a line in-between existing slides.
- 

- **Step 3** – When you get to the right position release the left click button to insert the slide. Alternately you can also cut the selected slide and paste it back in the sequence as shown below.

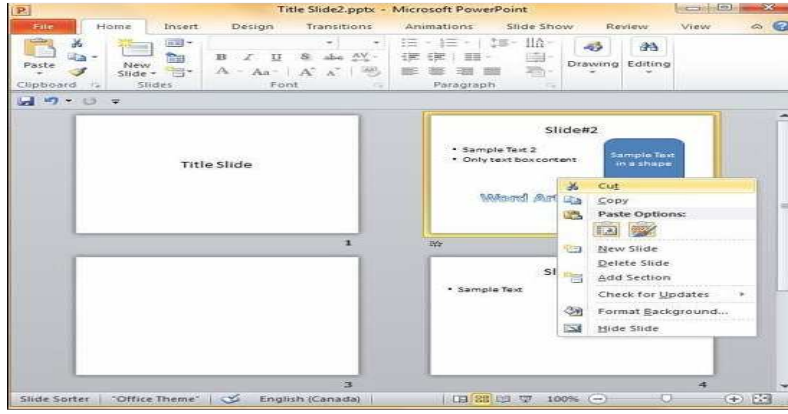


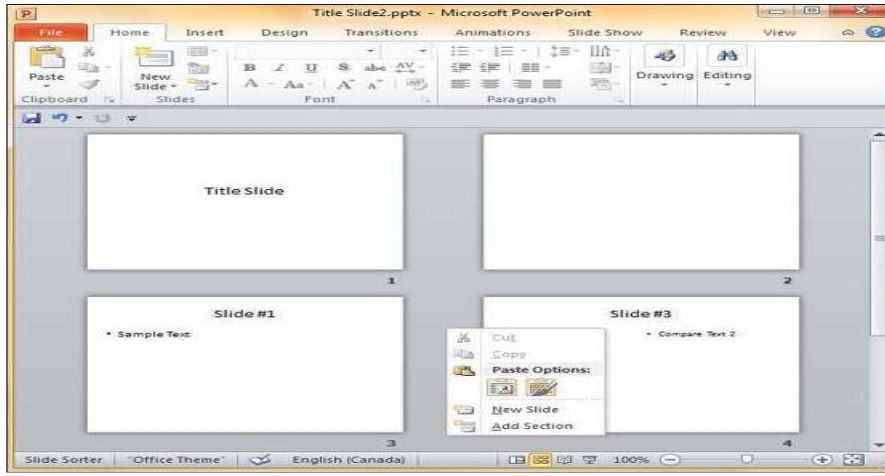
- Slide Sorter View

- Let us now understand how the Slide Sorter View works.
- **Step 1** – Select the slide to be moved.
- **Step 2** – Left click on the slide and drag it to the position in the sequence where you want to place it. PowerPoint will indicate the insert position with a line in-between existing slides.



- **Step 3** – When you get to the right position release the left click button to insert the slide. Alternately you can also cut the selected slide and paste it back in the sequence as shown below.

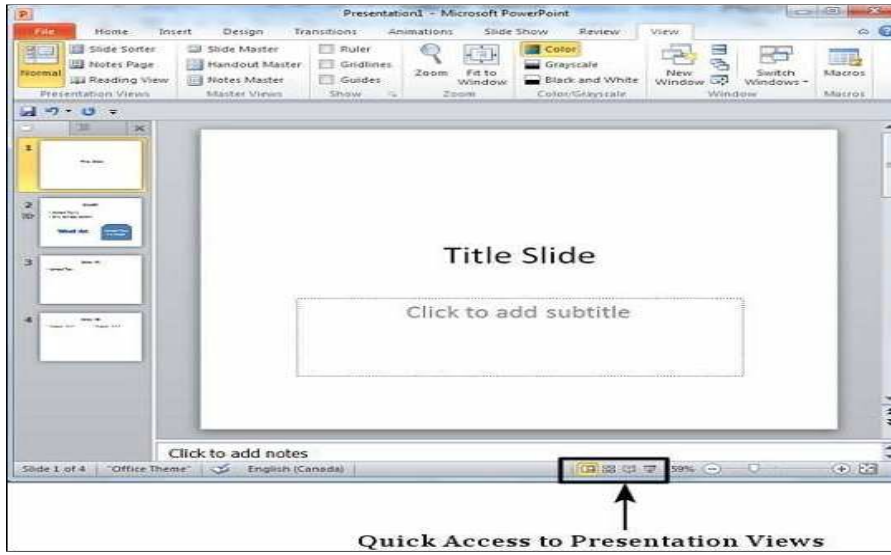




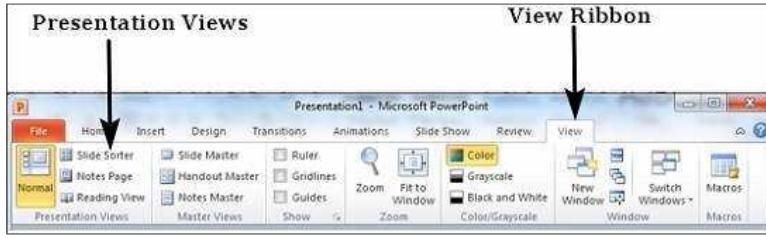
PowerPoint supports multiple views to allow users to gain the maximum from the features available in the program. Each view supports a different set of functions and is designed accordingly.

PowerPoint views can be accessed from two locations.

- Views can be accessed quickly from the bottom bar just to the left of the zoom settings.



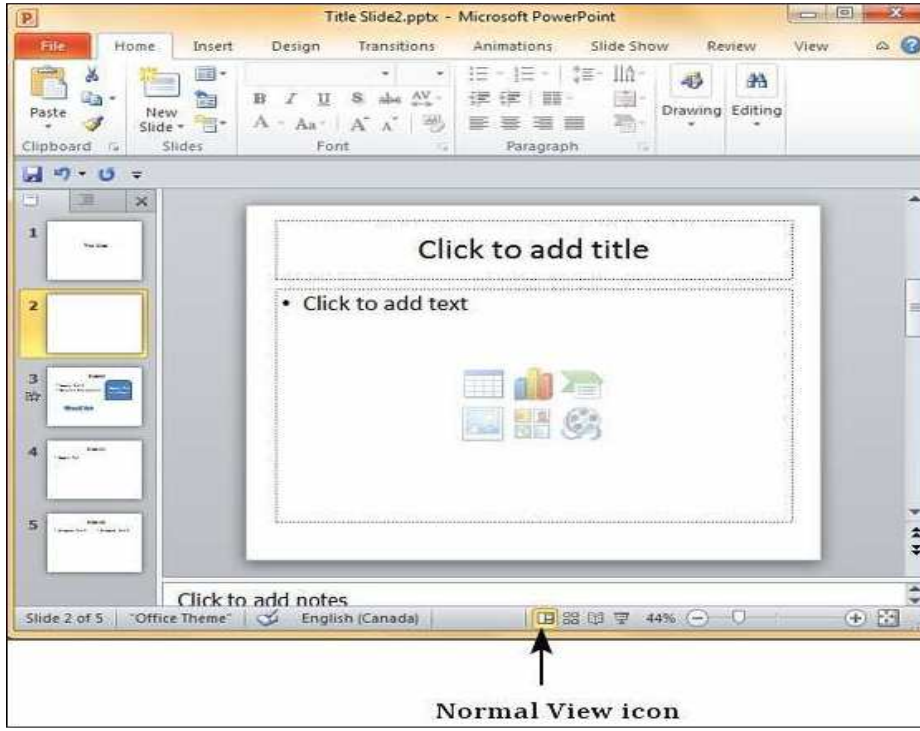
- Views can also be accessed from the **Presentation Views** section in the View ribbon



Here is a short description of the various views and their features.

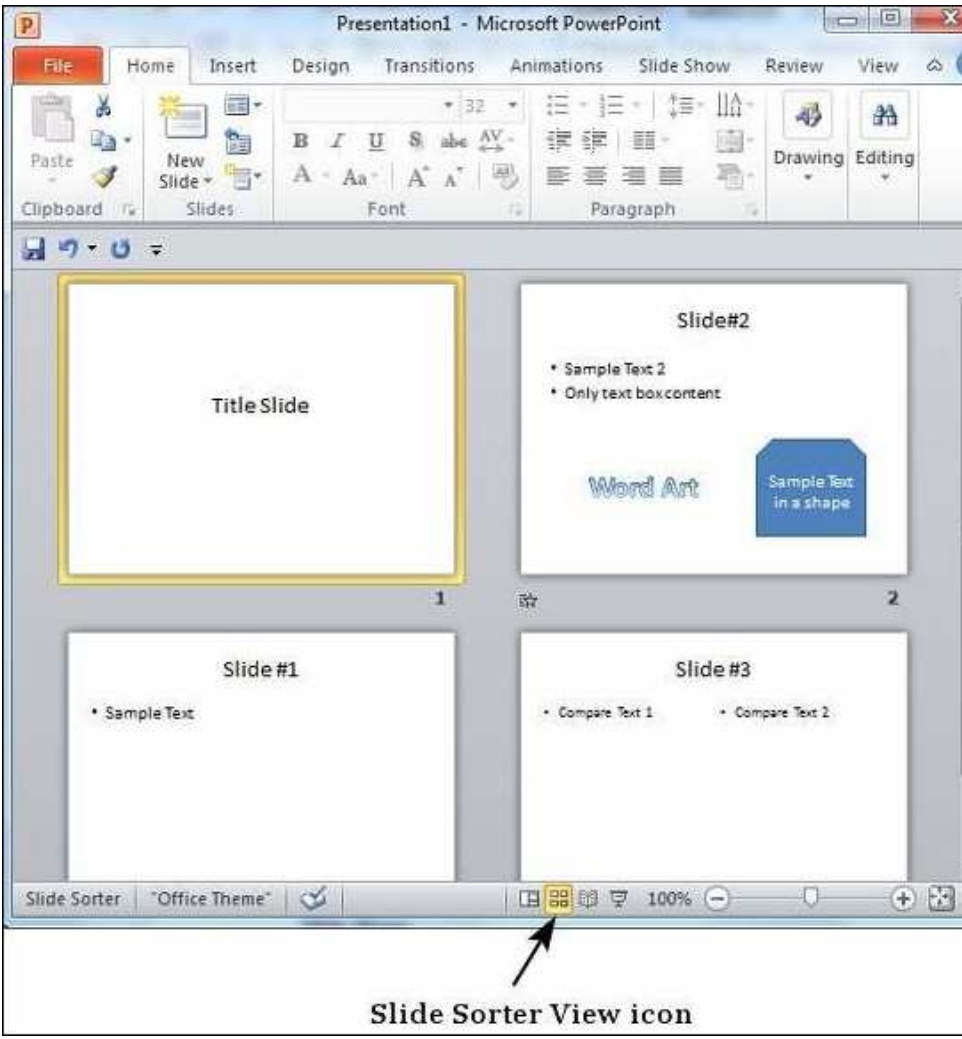
Normal View

This is the default view in PowerPoint and this is primarily used to create and edit slides. You can create/ delete/ edit/ rearrange slides, add/ remove/ modify content and manipulate sections from this view.



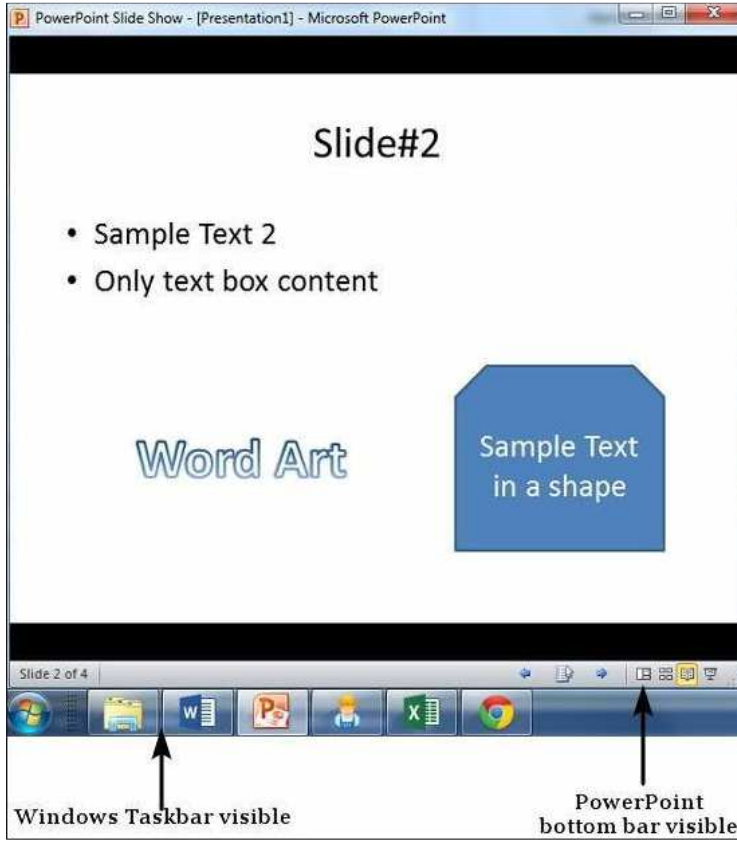
Slide Sorter View

This view is primarily used to sort slides and rearrange them. This view is also ideal to add or remove sections as it presents the slides in a more compact manner making it easier to rearrange them.



Reading View

This view is new to PowerPoint 2010 and it was created mainly to review the slideshow without losing access to rest of the Windows applications. Typically, when you run the slideshow, the presentation takes up the entire screen so other applications cannot be accessed from the taskbar. In the reading view the taskbar is still available while viewing the slideshow which is convenient. You cannot make any modifications when on this view.



SlidesShow

This is the traditional slideshow view available in all the earlier versions of PowerPoint. This view is used to run the slideshow during presentation.