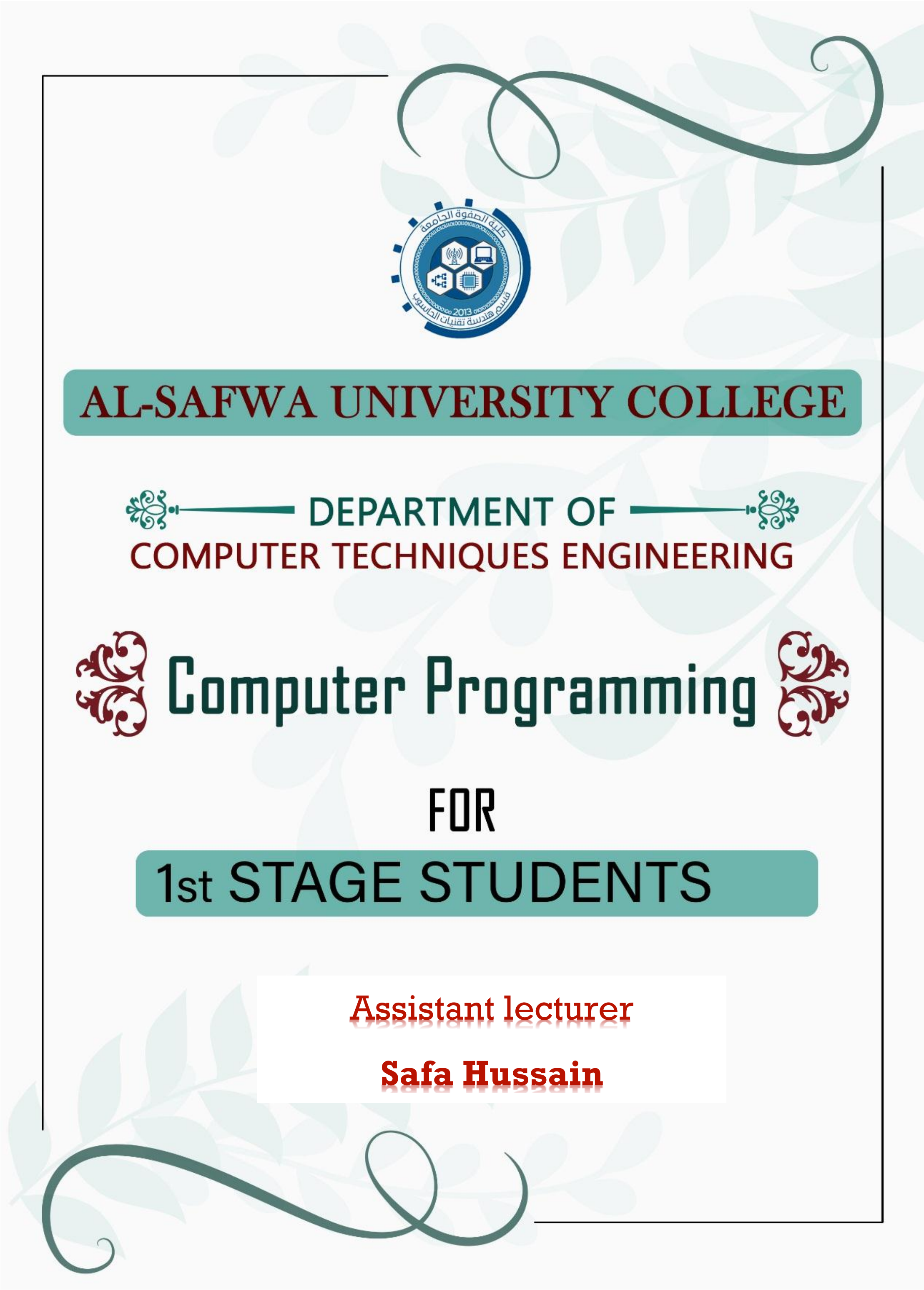


AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

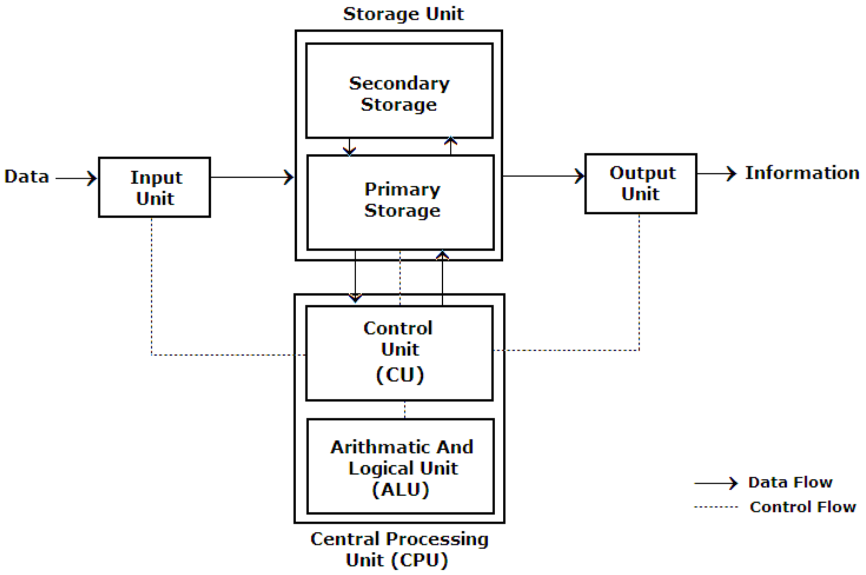
1st STAGE STUDENTS

Assistant lecturer

Safa Hussain

What is the computer:

0The computer is a device that operated upon information or data. It is an electronic device which accepts inputs data, stores the data, does arithmetic and logic operations and provides the outputs in the desired format. The computer receives data, process it, produces output and stores it for further references. So, a computer should have at least four major components to perform these tasks. A block diagram of the basic computer organization has the following functional units.



Four logical units in every computer:

1.**Input unit**

•Obtains information from input devices (keyboard, mouse)

2.Output unit

•Outputs information (to screen, to printer, to control other devices)

3.Memory unit

•RAM

–Quick access, low capacity, stores input information

•Secondary storage unit

–Cheap, long-term, high-capacity storage

–Stores inactive programs

4 .Central processing unit (CPU)

–Supervise and coordinate the other sections of the computer

•Arithmetic logic unit (ALU)

–Performs arithmetic calculations and logic decisions

•Control Unit (CU)

– The control unit is like a traffic warden directing and controlling the flow of data

COMPUTER SYSTEM

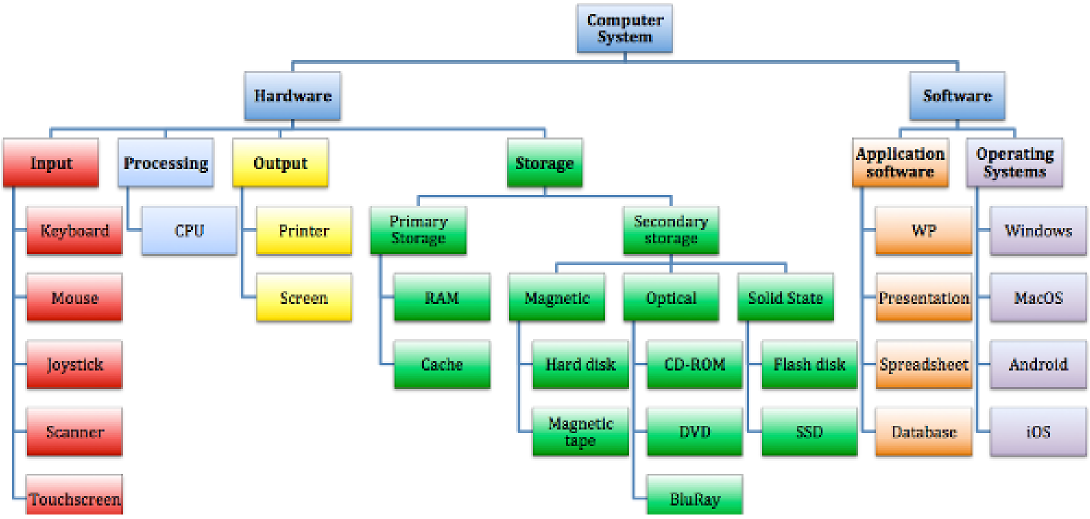
Is a collection of entities(hardware,software) that are designed to receive, process, manage and present information in a meaningful format.

COMPONENTS OF COMPUTER SYSTEM

- **Computer hardware** - Are physical parts/ intangible parts of a computer. eg Input devices, output devices, central processing unit and storage devices
- **Computer software** - also known as programs or applications. They are classified into two classes namely - sytem software and application software

ITGS – Hardware and Software overview

Complete the chart below by filling in the gaps



Programming:

A set of commands that are given to the computer to perform a specific work in a way that the computer understands. For example, give the computer a command to print a sentence or run a calculation in a way that the computer calculates.

How can I communicate with the computer?

In a computer-understood language that is electrical signals (1-0).

How can I communicate with the computer and the programmer?

Through a set of programming languages.

Programming languages:

A programming language is a set of rules that provides a way of telling a computer what operations to perform. A programming language is a set of rules for communicating an algorithm. It provides a linguistic framework for describing computations**.**

In other term:

Is a set of tools that provide written commands in a language understood by the programmer and converted to the machine language using a set of tools and programs.

Three types of programming languages :

- ❖ **Machine languages**

  machine language is the binary language made of 1s and 0s is executed directly by a computer. It is difficult, dependent on the type of machine. Difficult to correct the errors

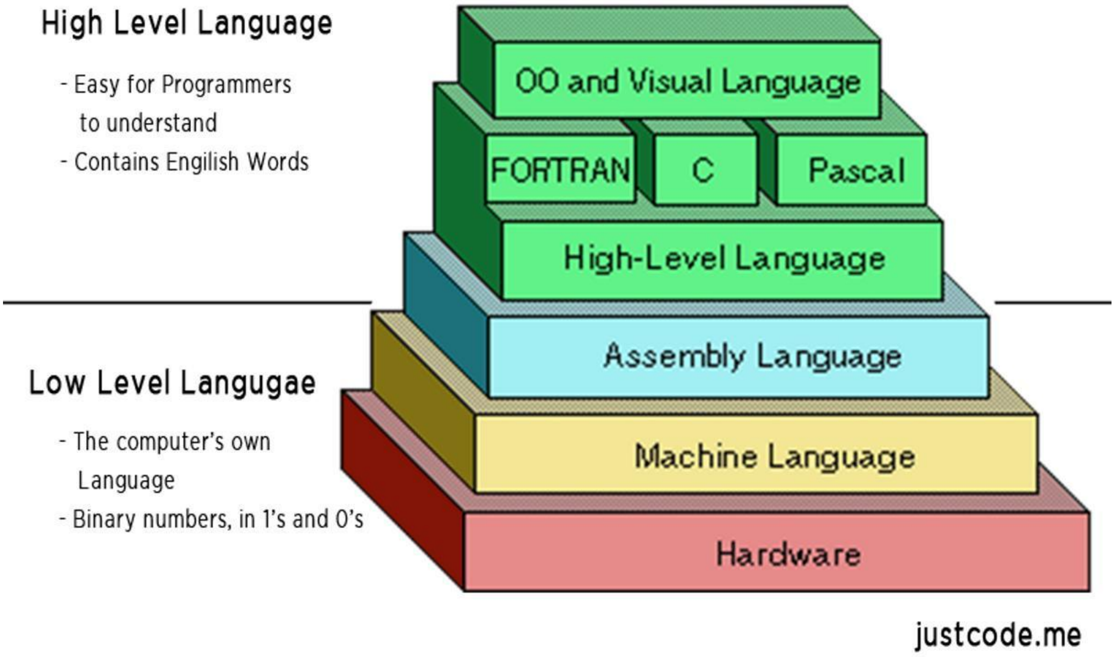
- ❖ **Assembly languages**

  To solve the above problems An **Assembly language** is design . It is a little easier than machine language, It uses symbols, and abbreviations to describe the instruction for example add, sub, jump. An **Assembly language** is called *symbolic machine code*.

Assembly language is converted into executable machine code by a utility program referred to as an *assembler*. The conversion process is referred to as *assembly*, or *assembling* the source code.

- **High-level languages**

  high-level language is a programming language such as C, FORTRAN, or Pascal that enables a programmer to write programs that are more or less independent of a particular type of computer. Such languages are considered high-level because they are closer to human languages and further from machine languages.



What is c++?

C++ (1983)(by Bjarne Stroustrup).

C++ is a general-purpose object-oriented programming (OOP) language, developed by Bjarne Stroustrup, and is an extension of the C language. It is therefore possible to code C++ in a "C style" or "object-oriented style." In certain scenarios, it can be coded in either way and is thus an effective example of a hybrid language.

C++ is considered to be an intermediate-level language, as it encapsulates both high- and low-level language features. Initially, the language was called "C with classes" as it had all the properties of the C language with an additional concept of "classes." However, it was renamed C++ in 1983.

1. It supports all features of both structured programming and OOP.

2. C++ focuses on function and class templates for handling data

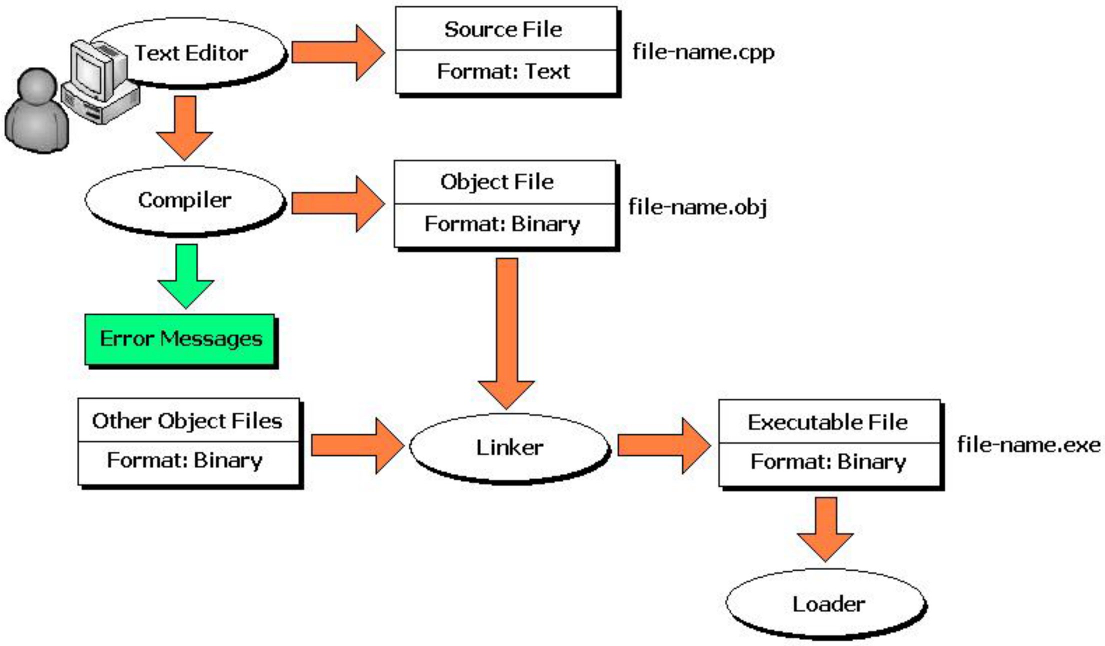
Types.

C++ Program Development Process (PDP):

C++ programs typically go through six phases before they can be executed. These phases are:

1. Edit: The programmer types a C++ source program, and makes correction, if necessary. Then file is stored in disk with extension (.cpp).
2. Pre-Processor: Pre-processing is accomplished by the pre-proceccor before compilation, which includes some substitution of files and other directories to be include with the source file.
3. Compilation: Converting the source program into object-code.
4. Linking: A linker combines the original code with library functions to produce an executable code.
5. Loading: The loader loads the program from the disk into memory.
6. CPU: Executes the program, residing in memory.

These steps are introduced in the figure below:



Algorithm:

algorithm can be defined as a finite sequence of effect statements to solve a problem. An effective statement is a clear, unambiguous instruction that can be carried out.

Algorithm properties:

•**Finiteness:** the algorithm must terminate a finite number of steps.

• **Non-ambiguity:** each step must be precisely defined. At the completion of each step the nest step should be uniquely determined .

•**Effectiveness:** the algorithm should solve the problem in a reasonable amount of time.

Ex1:Write algorithm to find maximum number from two numbers.

```
1.start
2. input two numbers(num1 and num2)
3.if num1 greater than num2 print num1
4.if num2 greater than num1 print num2
5. end
```

Ex2: Write algorithm to sum two numbers

1.Start

2.Input first number

3.Input second number

4.Calculate the sum

5.Print the result

6.end

Ex3 Write algorithm to find the division two number

1.Start

2.Input first number

3.Input second number

4.If second number <>0 Goto 6

5.If second number=0 Goto 9

6.Calculate division

7.Print the result

8.Goto 10

9.Print error

10.end

Ex 4:Write algorithm o find the sum of negative numbers among 50 numbers.

1.Start

2.Initialize two variables (count and sum), set count to 0 ,sum to 0

3.Input number.

4. increment the counter of input numbers

5.If the number is negative go to 7

6.If the number is positive go to 8

7.Add the number to the sum.

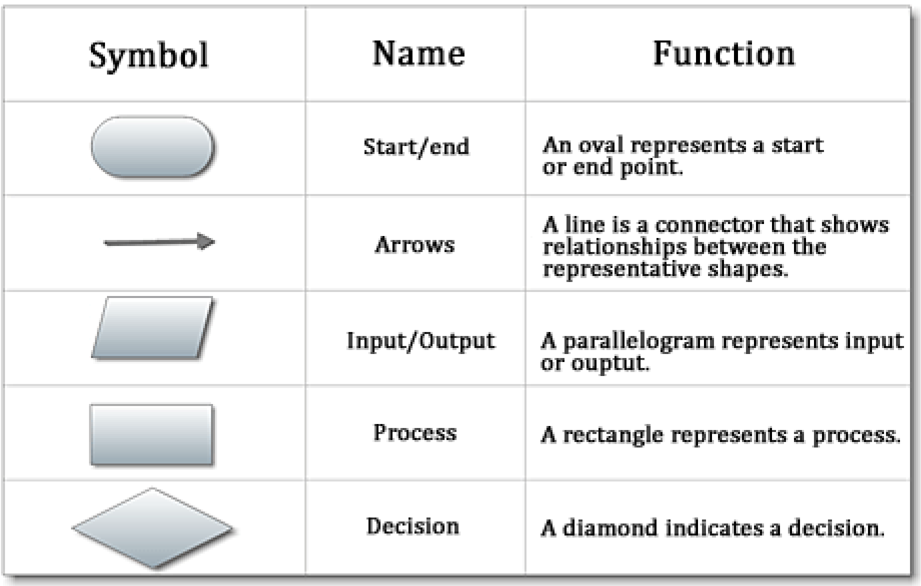
8.if the counter less than 50 go to 3

9. print sum

10. End

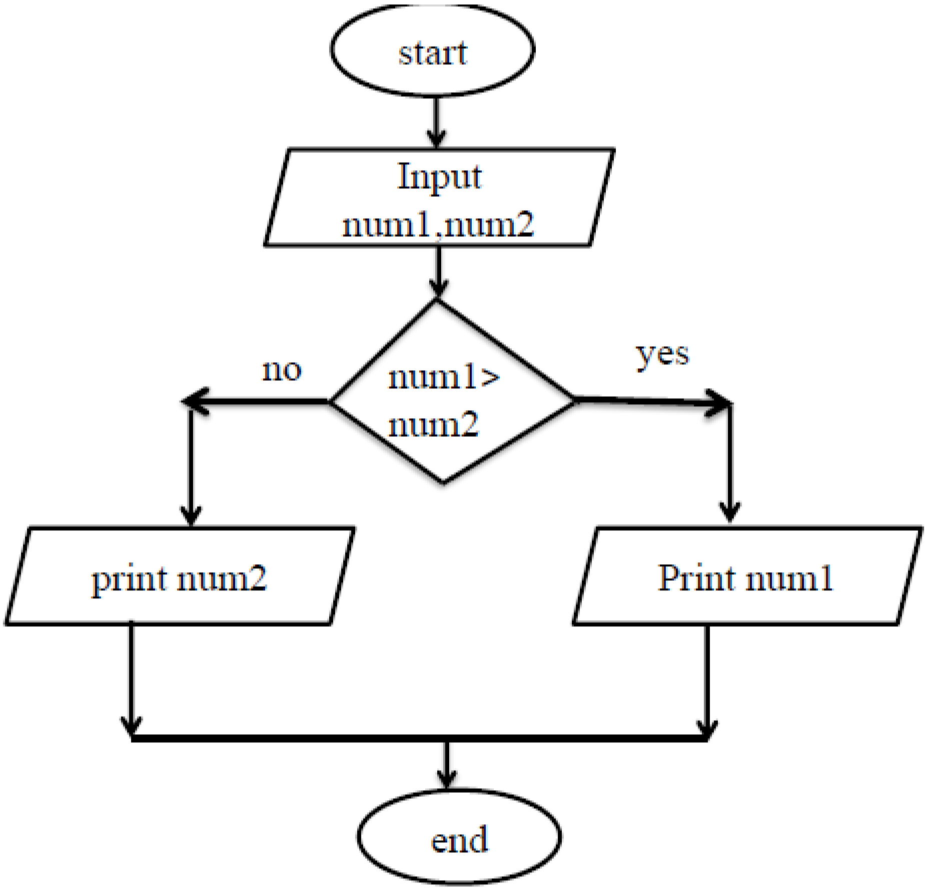
A flowchart

is a graphical representation of an algorithm or of a portion of an algorithm .Flowcharts are drawn using symbols. The main symbols used to draw a flowchart are shown in following figure .

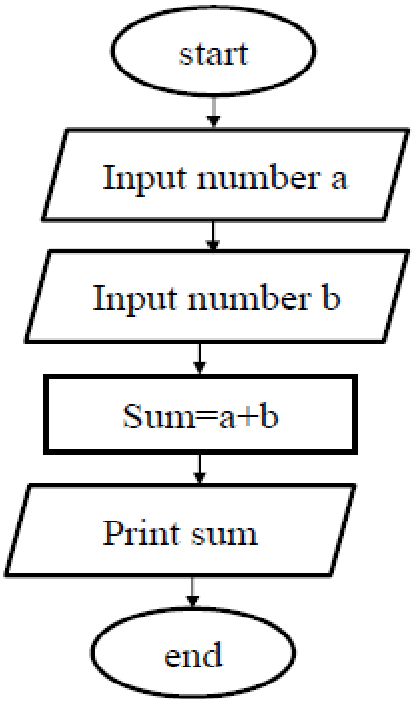


Symbol	Name	Function
⬭	Start/end	An oval represents a start or end point.
→	Arrows	A line is a connector that shows relationships between the representative shapes.
▱	Input/Output	A parallelogram represents input or ouptut.
▭	Process	A rectangle represents a process.
◇	Decision	A diamond indicates a decision.

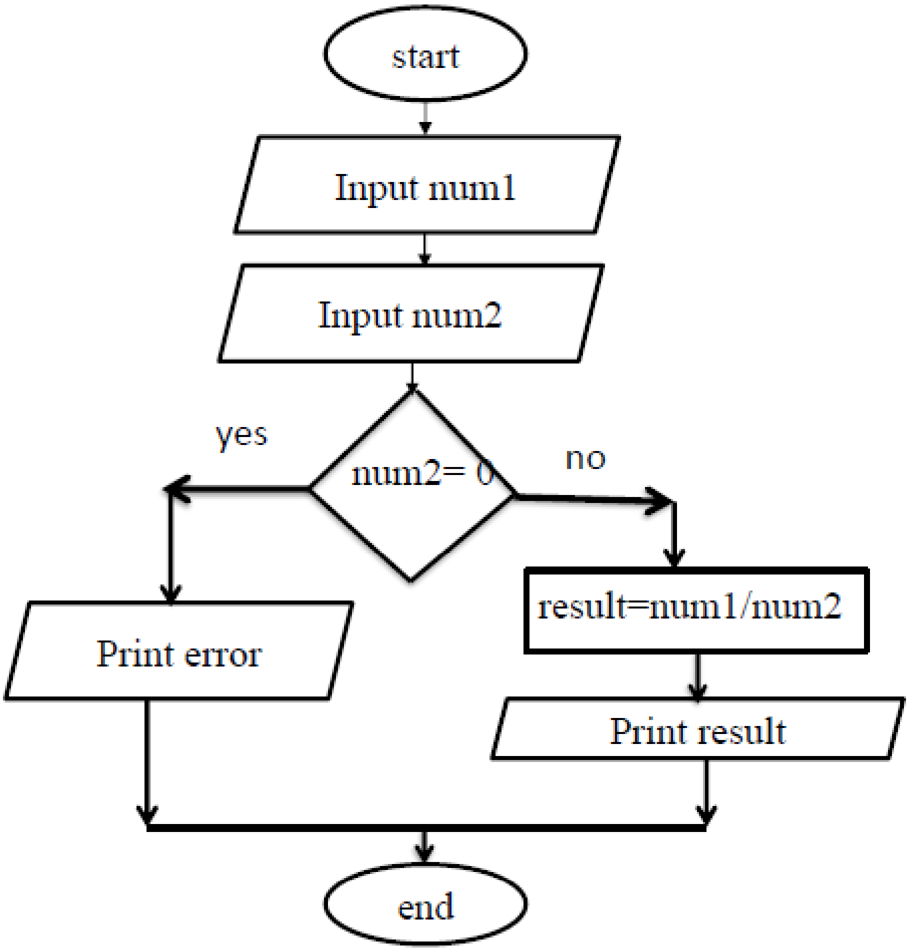
Ex1:Draw the flowchart of maximum number between two numbers?



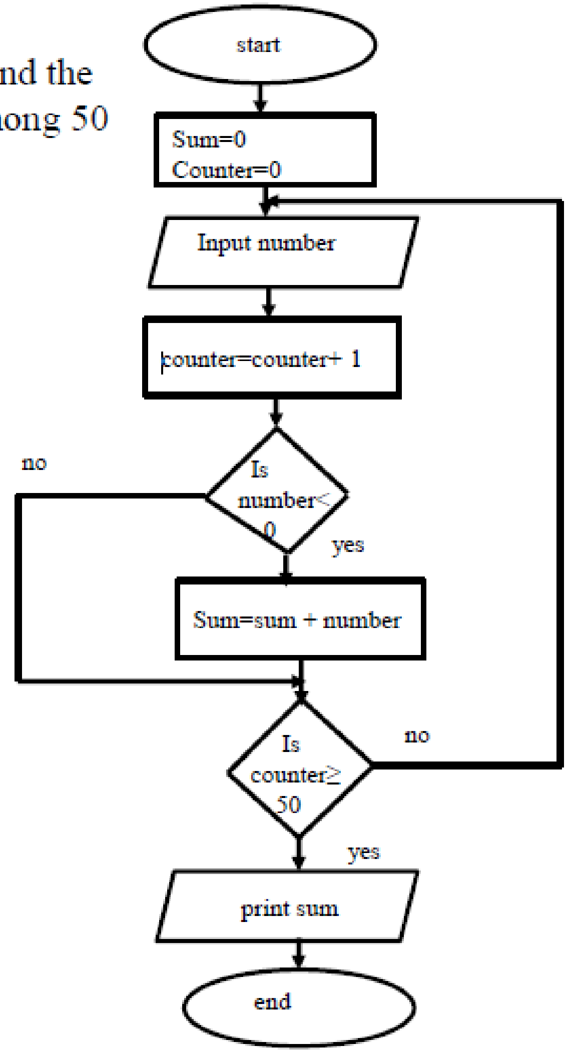
Ex2: draw the flowchart of sum two numbers?



Ex3: draw the flowchart of the division of two numbers?



ex4 :draw the flowchart to find the sum of negative numbers among 50 numbers.



Homework:

1)Draw a flowchart to read 3 numbers: x , y and z and print the largest number of them.

2)Write an algorithm that finds the average of three degrees. If the avarage is successful, printed word "success" ,opposite of that printed word"fail"?

AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

1st STAGE STUDENTS

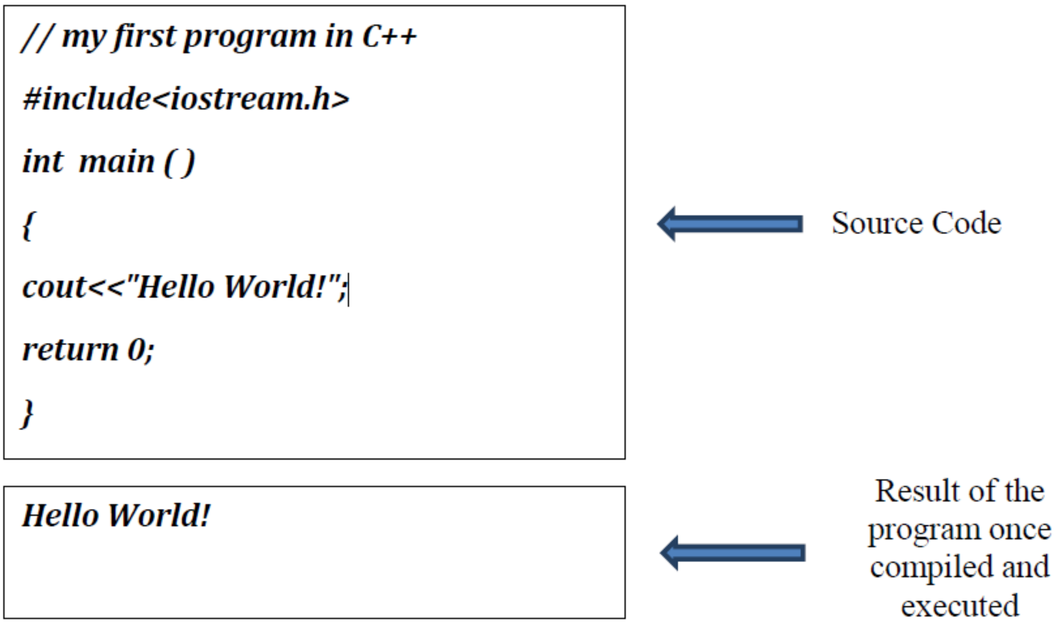
LECTURER
MSc.SAFA HUSSAIN

Introduction to c++

Structure of aprogram

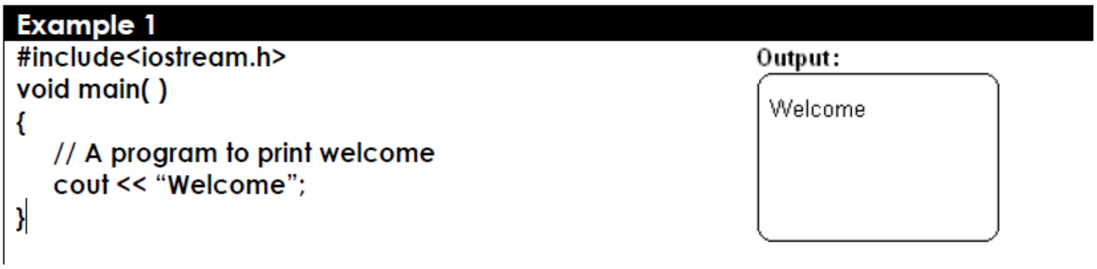
Stages of building the program

1.Write comments.
2.Definition of variables.
3.Input variables
4.Processing.
5-Take out the results.



```
// my first program in C++
#include<iostream.h>
int  main ( )
{
cout<<"Hello World!";
return 0;
}
```

```
Hello World!
```



Example 1

```
#include<iostream.h>
void main( )
{
    // A program to print welcome
    cout << “Welcome”;
}
```

Output:

```
Welcome
```

// my first program in C++

This is a comment line. All lines beginning with two slash signs (//) are considered comments and do not have any effect on the behavior of the program

#include<iostream.h>

Lines beginning with a hash sign (#) are directives for the preprocessor. In this case the directive #include <iostream> tells the preprocessor to include the iostream standard file. This specific file (iostream) includes the declarations of the basic standard input-output library in C++, and it is included because its functionality is going to be used later in the program.

int main ()

This line corresponds to the beginning of the definition of the main function. The main function is the point by where all C++ programs start their execution, independently of its location within the source code. It is essential that all C++ programs have a main function. The word main is followed in the code by a pair of parentheses ( ). Right after these parentheses we can find the body of the main function enclosed in braces { }. What is contained within these braces is what the function does when it is executed

.

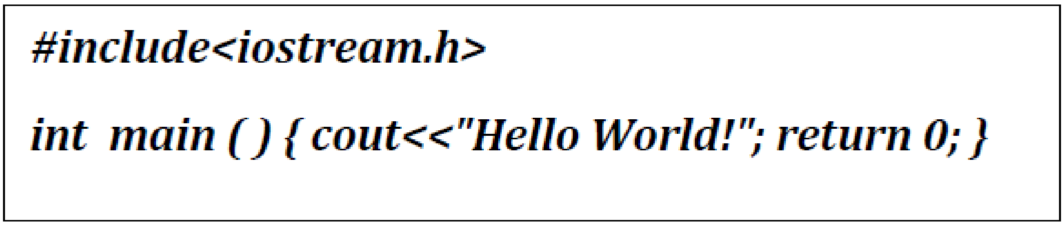
cout << "Hello World!";

This line is a C++ statement. A statement is a simple or compound expression that can actually produce some effect. In fact, this statement performs the only action that generates a visible effect in our first program. cout represents the standard output stream in C++. Notice that the statement ends with a semicolon character (;). This character is used to mark the end of the statement and in fact it must be included at the end of all expression statements in all C++ programs

.

return 0;

The return statement causes the main function to finish.
The two lines code int main () and return 0; can be replaced by only the following line code:
void main ( )



```
#include<iostream.h>

int  main ( ) { cout<<"Hello World!"; return 0; }
```

Let us add an additional instruction to our first program:

```
// my second program in C++
#include<iostream.h>
int  main ( )
{
cout<<"Hello World!";
cout<<" I'm a C++ program";
return 0;
}
```

```
Hello World! I'm a C++ program
```

Comments

Comments are parts of the source code disregarded by the compiler. They simply do nothing. Their purpose is only to allow the programmer to insert notes or descriptions embedded within the source code. C++ supports two ways to insert comments:

// line comment

Or

/* block comments */

We are going to add comments to our second program:

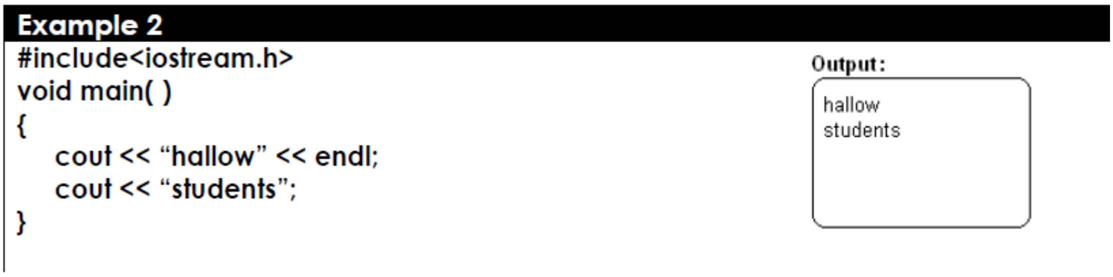
```
/* my second program in C++ with
More comments*/
#include<iostream.h>
int  main ( )
{
cout<< "Hello World!";//prints Hello World!
cout<<" I'm a C++ program";//prints I'm a C++ program
return 0;
}
```

Note:

{, introducing the statements that define the function.
}, indicates the end of the statements in the function.
//, text after these symbols is a comment. It does not affect the program code, and
compilers normally ignore it.

<<, the stream insertion operator (or send operator).
>>, the stream extraction operator (or get from operator).
; , semicolon, the terminator of every C++ statement.

The **endl** is used in c++ to represent a new line, as shown in the following example:



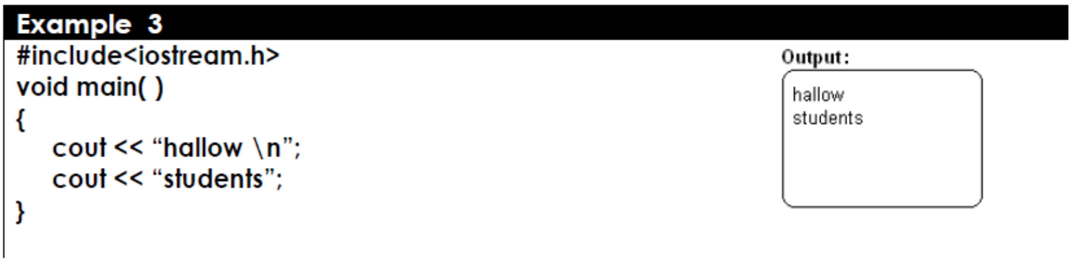
Example 2

```
#include<iostream.h>
void main( )
{
    cout << “hallow” << endl;
    cout << “students”;
}
```

Output:

```
hallow
students
```

The **\n** is a special escape code, also used in C++ to represent a new line, as shown in the following example:



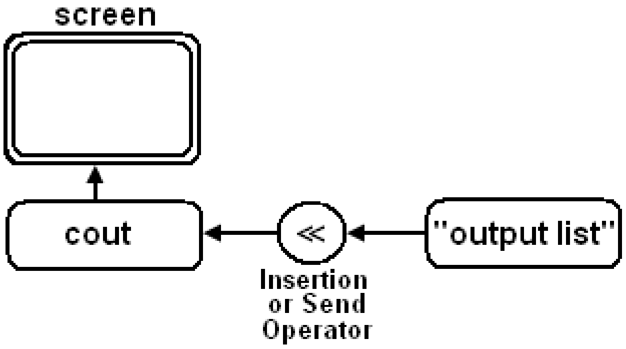
Example 3

```
#include<iostream.h>
void main( )
{
    cout << “hallow \n”;
    cout << “students”;
}
```

Output:

```
hallow
students
```

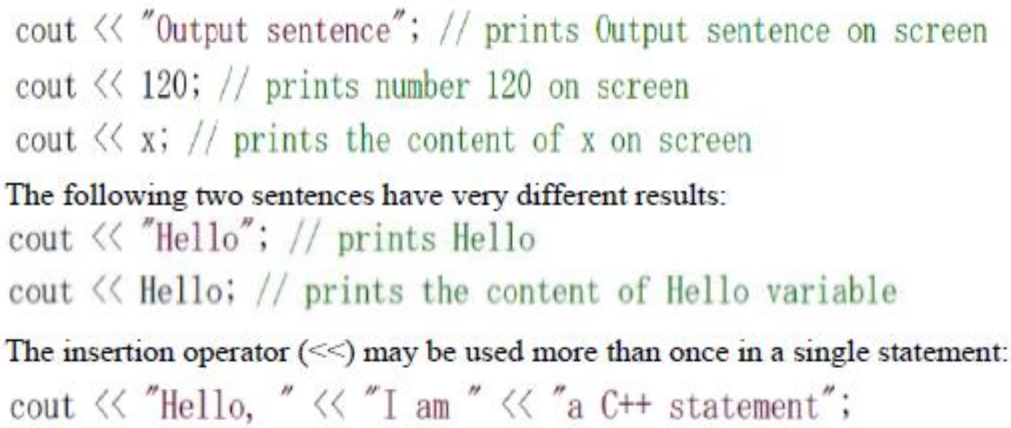
Standard Output (cout)



ex: cout<<"C++ Language"

By default, the standard output of a program is the screen, and the C++ stream object defined to access it is ***cout***<<.

For example:



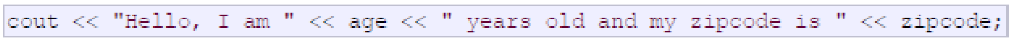
```
cout << "Output sentence"; // prints Output sentence on screen
cout << 120; // prints number 120 on screen
cout << x; // prints the content of x on screen
```

The following two sentences have very different results:

```
cout << "Hello"; // prints Hello
cout << Hello; // prints the content of Hello variable
```

The insertion operator (<<) may be used more than once in a single statement:

```
cout << "Hello, " << "I am " << "a C++ statement";
```



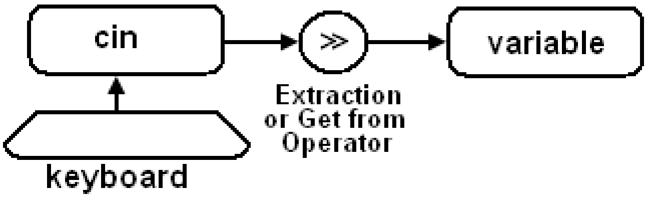
```
cout << "Hello, I am " << age << " years old and my zipcode is " << zipcode;
```

EX1: Write a C++ program that prints on screen the following sentences:

{****Alsafwah University College***}
{* Computer Techniques Engineering Department*}
{**First Year**}

```
#include <iostream.h>
void main ( )
{
cout << "{****Alsafwah University College***}\n";
cout << " {* Computer Techniques Engineering Department* }\n";
cout << " {**First Year**}";

}
```

Standard Input (cin)



<u>ex:</u> cin>>x;

The standard input device is usually the keyboard.. **For example**:

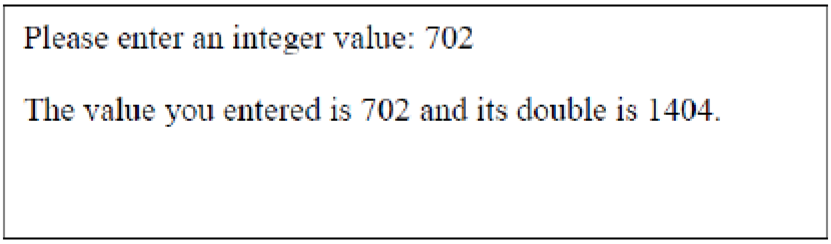
```
int age;
cin >> age;
```

You can also use ***cin*** to request more than one datum input from the user:
cin>>a>>b;

is equivalent to :

cin>>a;
cin>>b;

EX2: Write a C++ program that inter any integer value and find its double, the screen output should be like:



```
Please enter an integer value: 702

The value you entered is 702 and its double is 1404.
```

```
#include <iostream.h>
int main ( )
{
int i;
cout << "Please enter an integer value: ";
cin >> i;
cout << "The value you entered is " << i;
cout << " and its double is " << i*2;
return 0;
}
```

Character set:

C++ has the letters and digits, as show below:
Uppercase: A, B, C, . . ., Z
Lowercase: a, b, c, . . ., z
Digits: 0, 1, 2, . . .,9

Special Characters: All characters other than listed treated as special characters for example

+ - * / ^

([{ }]

) < = > , (Comma)

“ (Double Conations) . (Dot) : (Colon) ; (Semicolon)
(Blank Space)

Note:

In C++ language, upper case and lower case letters are distinct and hence there are 52 letters in all. For example **bag** is different from **Bag** which is different from **BAG**

Identifiers:

An identifier is a name given to some program entity, such as variable, constant, array, function, structure, or class.

1-An identifier is a sequence of alphanumeric (alphabetic and numeric) characters
2- The identifier can be lowercase or uppercase letters not affect
3-, Identifier must not contain spaces.
4-A variable should not begin with a digit in the first chaeacter ,but(in middle or end the name no effect).
5- C++ does not set a maximumlength for an identifier.

Some examples of valid identifiers are as follows:

MY_NAME, and, and_eq, bitand, bitor, compl, not,

not_eq, or, or_eq, xor, xor_eq, zadd, b_add,

sum, add2num, First_Num

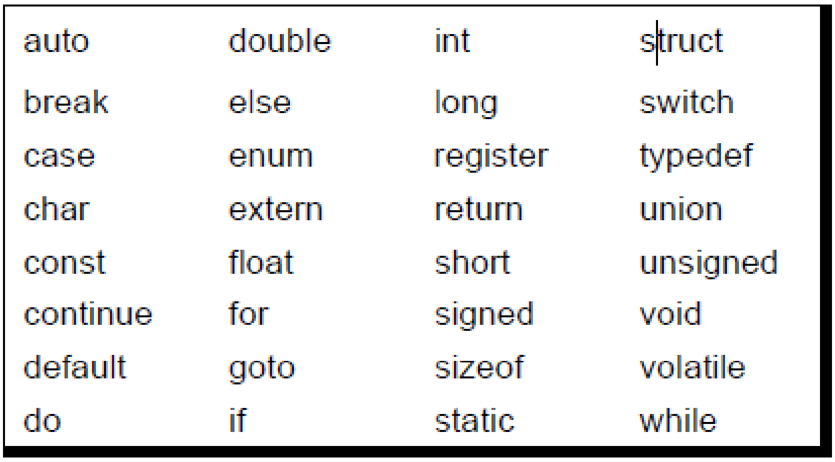
Examples of invalid identifiers are:

3ab , a()test , ros sal , tan*x, sin-x, void,

f(x), s%t, s^a , S1.1, d1,d , f/r

Keywords:

The keywords are also identifiers but cannot be user defined, since they are reserved words. All the keywords should be in lower case letters. Reserved words cannot be used as variable names or constant. The following words are reserved for use as keywords:



auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Constants and variable

Constants are expressions with a fixed value.

Such us

a = 5;

the **5** in this piece of code was a literal constant.Literal constants can be divided in **Integer Numerals, Floating-Point Numerals, Characters, Strings and Boolean Values.**

Integer Numbers:

For examble:

1776

707

-273

Floating Point Numbers

For examble:

6.02e23f // float

Character and string literals

There also exist non-numerical constants, like:
'z'
'p'
"Hello world"
"How do you do?"

Note:

The first two expressions represent single character constants, and the following two represent string literals

composed of several characters. Notice that to represent a single character we enclose it between single quotes (')

and to express a string (which generally consists of more than one character) we enclose it between double quotes

(").

Symbolic constant

The declaration of constant value variables will be according to the following formula:

const Variable type Variable name= The value of the variable

Example:

Pi=3.141593?
Const double pi=3.141593;

Note:the value of symbolic constant cannot be changed.

Example 6

Write a program that reads the radius of a circle, then computes and outputs its area.

```
#include<iostream.h>
void main( )
{
    const float  pi = 3.14;
    int r;  float c;
    cout << “enter the radius of circle:”;
    cin>>r;
    cout<<endl;
    c = r * r * pi;
    cout << “the area of circle:” << c;
}
```

Output:

```
enter the radius of circle: 5

the area of circle: 78.5
```

Example 7

The following program computes the arethmatic operators.

```
#include<iostream.h>
void main( )
{
    int a,b,sum,sub,mul,div;
    cout << “enter any two numbers<<endl;
    cin>> a>>b;
    sum=a+b;
    sub=a-b;
   mul=a*b;
   div=a/b;
   cout<<”a=”<<a<<”b=”<<b<<”sum=”<<sum<<endl;
   cout<<”sub=”<<sub<<endl;
   cout<<”mul=”<<mul<<endl;
   cout<<”div=”<<div<<endl;
}
```

Output:

```
Enter any two numbers
10    20
A=10  b=20  sum=30
Sub=-10
Mul=200
Div=0
```

Example 8

The following program computes different division operators.

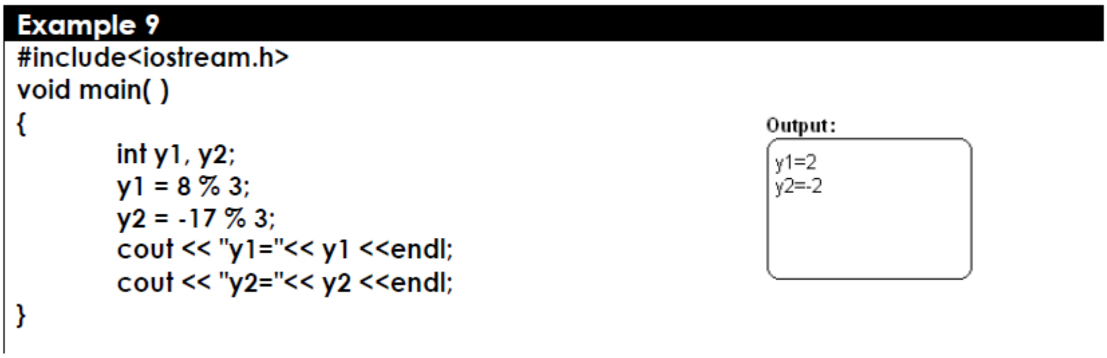
```
#include<iostream.h>
void main( )
{
            int x,  y, z, r ;
            x= 7 / 2;
            cout << "x=" << x <<endl;
            y=17/(-3);
            cout << "y="<< y <<endl;
            z=-17/3;
            cout << "z="<< z <<endl;
            r=-17/(-3);
            cout << "r="<< r <<endl;
}
```

Output:

```
x= 3
y= -5
z= -5
r= 5
```

Note:

The modulus operator **“%”** is used with **integer** operands (int, short, long, unsigned). It **can’t** be used with **float or double** operands.



Example 9

```
#include<iostream.h>
void main( )
{
        int y1, y2;
        y1 = 8 % 3;
        y2 = -17 % 3;
        cout << "y1="<< y1 <<endl;
        cout << "y2="<< y2 <<endl;
}
```

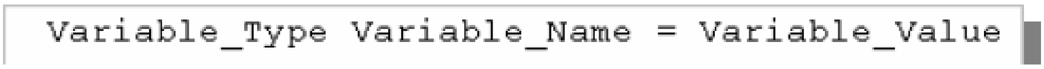
Output:

```
y1=2
y2=-2
```

Variables

Variable is an entity that may vary during program execution. Variable names are names given to locations in memory. These locations can contain variable value.

Variables Declaration



```
Variable_Type Variable_Name = Variable_Value
```

Rules for Constructing Variable Names (**identifier**)
** The same conditions are mentioned in(**identifier**)

Initialization of variables
When declaring a regular local variable, its value is by default undetermined. But you may want a variable to store
a concrete value at the same moment that it is declared. In order to do that, you can initialize the variable. There
are two ways to do this in C++:

1-type identifier = initial_value ;
For example,
if we want to declare an int variable called a initialized with a value of 0 at the moment in which it is declared, we could write:

int a = 0;

2-The other way to initialize variables, known as constructor initialization, is done by enclosing the initial value
between parentheses **(()):**

type identifier (initial_value) ;

For example:

```
int a (0);
```

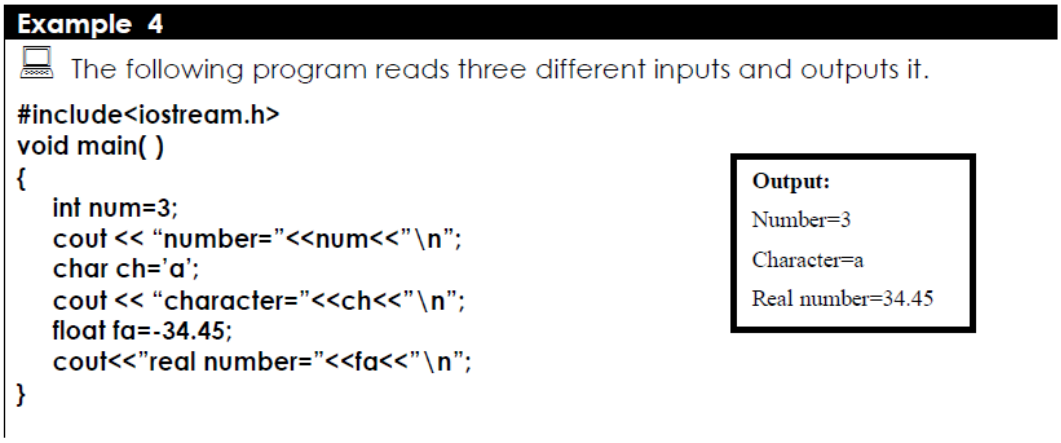
Both ways of initializing variables are valid and equivalent in C++

Examble1:

```
// initialization of variables
#include <iostream>
int main ()
{
int a=5; // initial value = 5
int b(2); // initial value = 2
int result; // initial value
undetermined
a = a + 3;
result = a - b;
cout << result;
return 0;

}
```

Examble2

```
// my first string
#include <iostream>
int main ()
{
string mystring = "This is a string";
cout << mystring;
return 0;
}
```



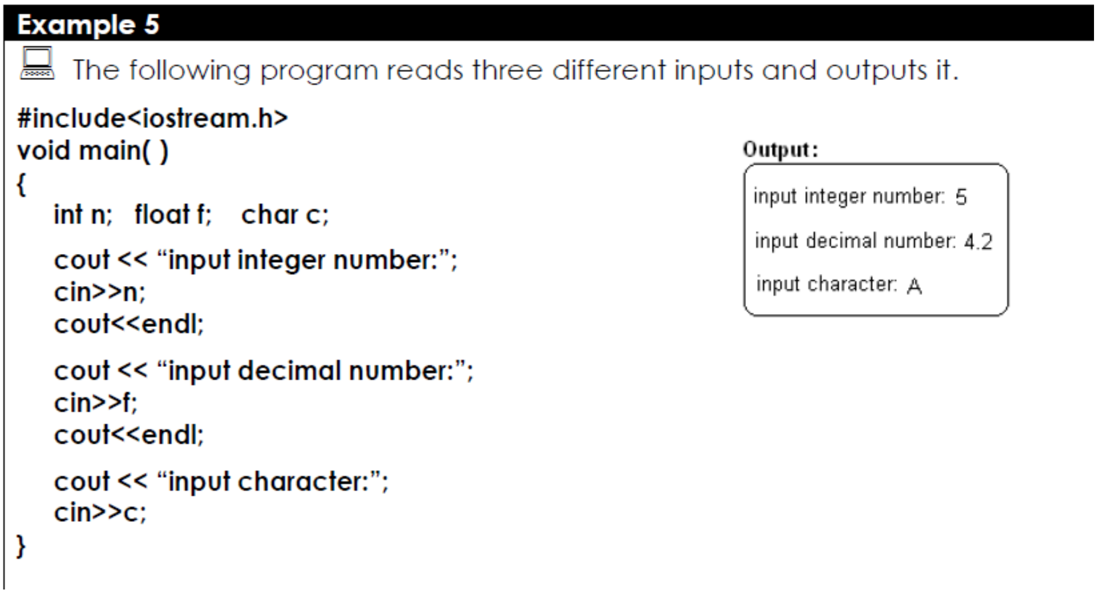
Example 4

The following program reads three different inputs and outputs it.

```
#include<iostream.h>
void main( )
{
    int num=3;
    cout << “number=”<<num<<”\n”;
    char ch=’a’;
    cout << “character=”<<ch<<”\n”;
    float fa=-34.45;
    cout<<”real number=”<<fa<<”\n”;
}
```

Output:

```
Number=3
Character=a
Real number=34.45
```



Example 5

The following program reads three different inputs and outputs it.

```
#include<iostream.h>
void main( )
{
    int n;   float f;   char c;

    cout << “input integer number:”;
    cin>>n;
    cout<<endl;

    cout << “input decimal number:”;
    cin>>f;
    cout<<endl;

    cout << “input character:”;
    cin>>c;
}
```

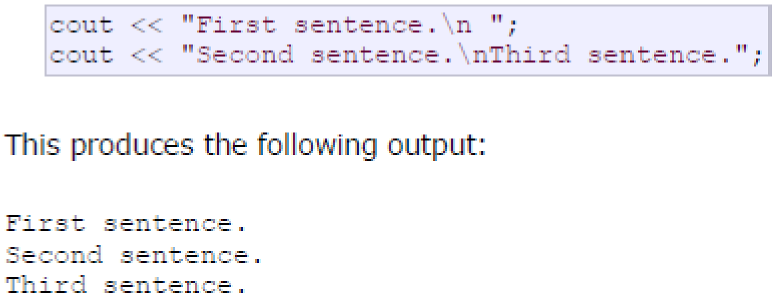
Output:

```
input integer number: 5
input decimal number: 4.2
input character: A
```

Special escape code

Escape code	description
\n	New line. Position the screen cursor to the beginning of the next line.
\t	Horizontal TAB . Move the screen cursor to the next tab stop.
\r	Carriage return. Position the cursor to the beginning of the current line, do not advance to the next line.
\b	Back space
\a	Alert. Produces the sound of the system bell.
\\	Backslash. Prints a backslash character.
\’	Single quote
\"	Double quote.
\?	question mark

Examble:



```
cout << "First sentence.\n ";
cout << "Second sentence.\nThird sentence.";
```

This produces the following output:

```
First sentence.
Second sentence.
Third sentence.
```

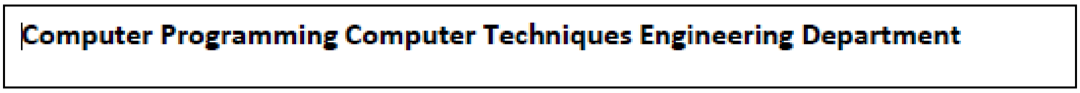
Additionally, to add a new-line, you may also use the endl manipulator. For example:
cout << "First sentence." << endl;
cout << "Second sentence." << endl;
would print out:
First sentence.
Second sentence.

Note:

It is important to notice that cout does not add a line break after its output unless we explicitly indicate it, therefore, the following statements:

cout<<"Computer Programming";
cout<<"Computer Techniques Engineering Department";

will be shown in the screen one following the other :



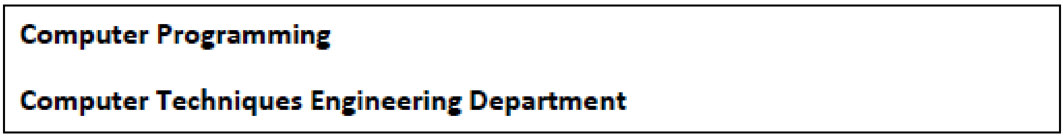
```
Computer Programming Computer Techniques Engineering Department
```

In C++ a new line character can be specified by using *\n* (new line) as follows:

cout<<"Computer Programming\n";
cout<<"Computer Techniques Engineering Department";

or <<*endl* (end line) as follows:

cout<<"Computer Programming "<<endl;
cout<<"Computer Techniques Engineering Department";



```
Computer Programming

Computer Techniques Engineering Department
```

Fundamental data types

When programming, we store the variables in our computer's memory, but the computer has to know what kind of

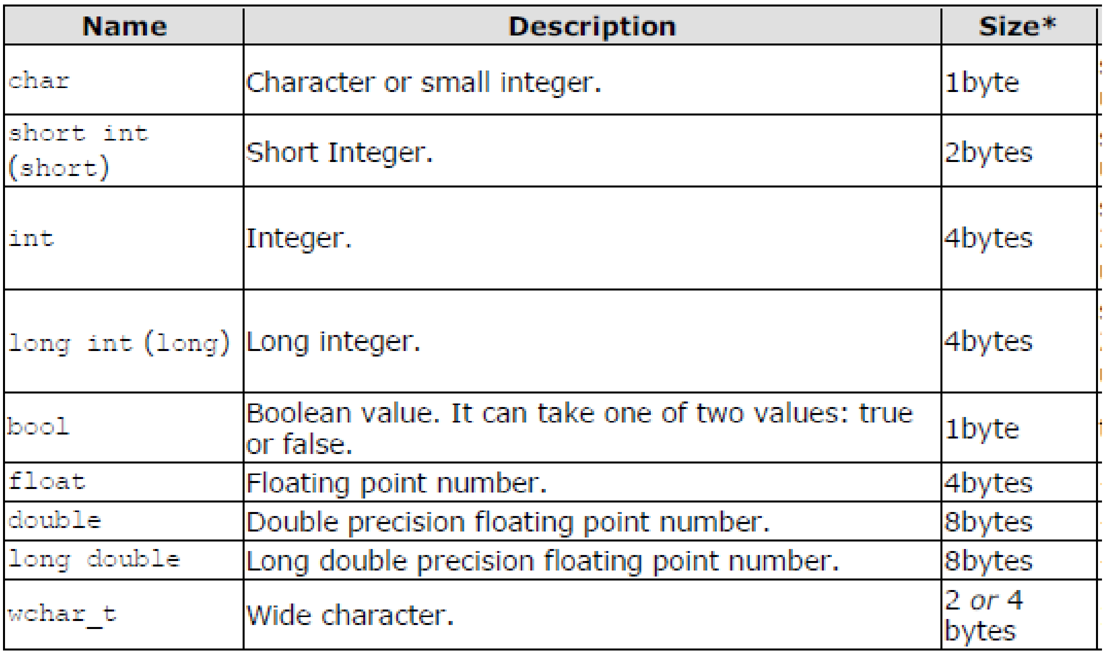
data we want to store in them, since it is not going to occupy the same amount of memory to store a simple

number than to store a single letter or a large number, and they are not going to be interpreted the same way.

The memory in our computers is organized in bytes. A byte is the minimum amount of memory that we can

manage in C++. A byte can store a relatively small amount of data: one single character or a small integer

(generally an integer between 0 and 255).



Name	Description	Size*
`char`	Character or small integer.	1byte
`short int (short)`	Short Integer.	2bytes
`int`	Integer.	4bytes
`long int (long)`	Long integer.	4bytes
`bool`	Boolean value. It can take one of two values: true or false.	1byte
`float`	Floating point number.	4bytes
`double`	Double precision floating point number.	8bytes
`long double`	Long double precision floating point number.	8bytes
`wchar_t`	Wide character.	2 *or* 4 bytes

Examble:

```
#include<iostream.h>
Int main()
{
Int s,h;
h=7;
s=8;
h=50.8;
cout<<h;
return(0);
}
```

Cout is 50 because declared h as an int.

AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

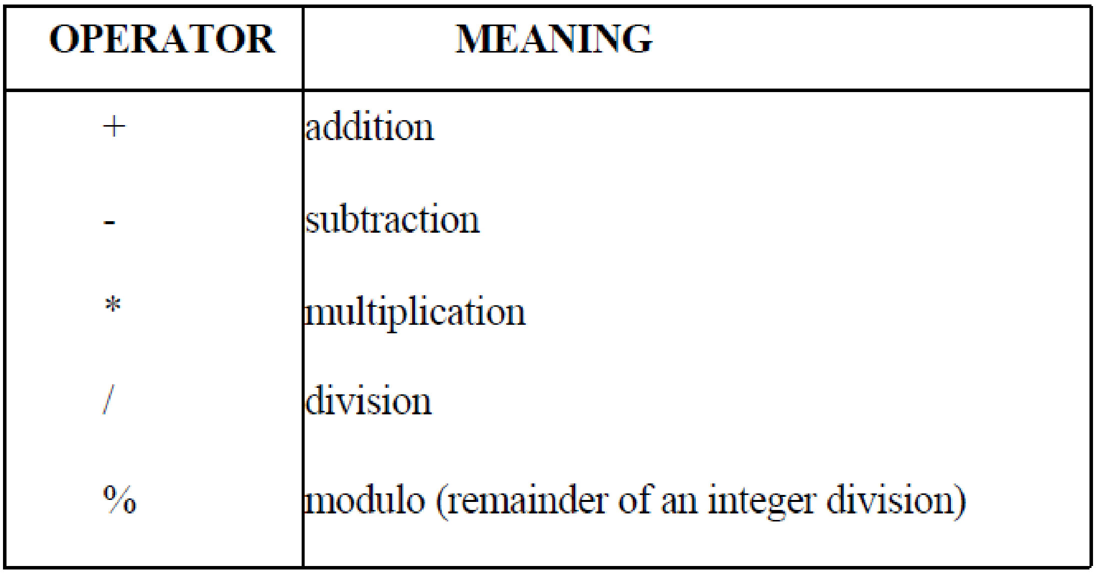
FOR

1st STAGE STUDENTS

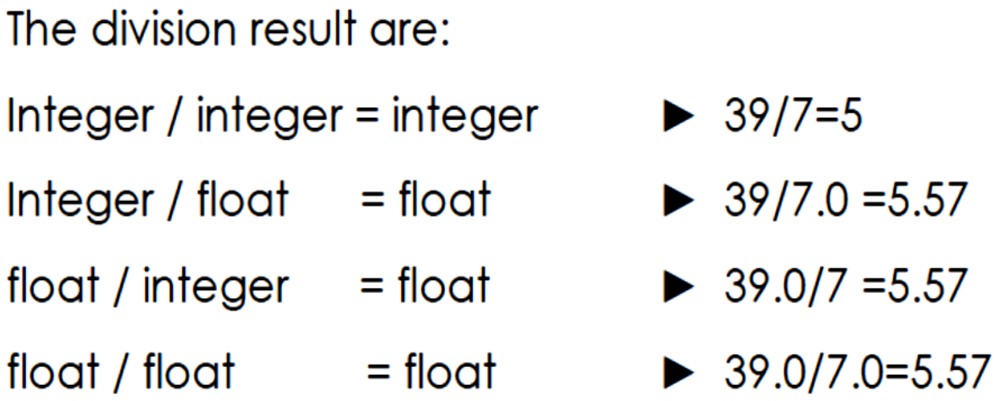
LECTURER
MSc.SAFA HUSSAIN

Arithmetic operators

Arithmetic operators: These operators require two variables to be evaluated:



OPERATOR	MEANING
+	addition
-	subtraction
*	multiplication
/	division
%	modulo (remainder of an integer division)



The division result are:

Integer / integer = integer ▶ 39/7=5

Integer / float = float ▶ 39/7.0 =5.57

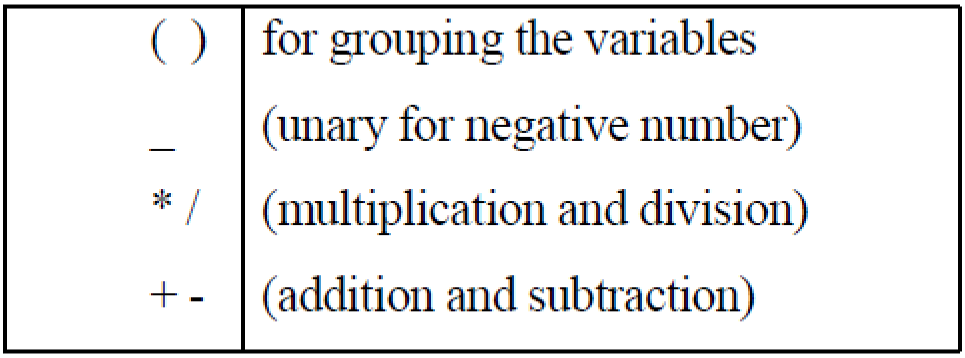
float / integer = float ▶ 39.0/7 =5.57

float / float = float ▶ 39.0/7.0=5.57

Example:

27%2=1 since 27=13*2+1 , 18%2=0 , while 39%5=4 since 39=7*5+4

Arithmetic operators as per precedence:

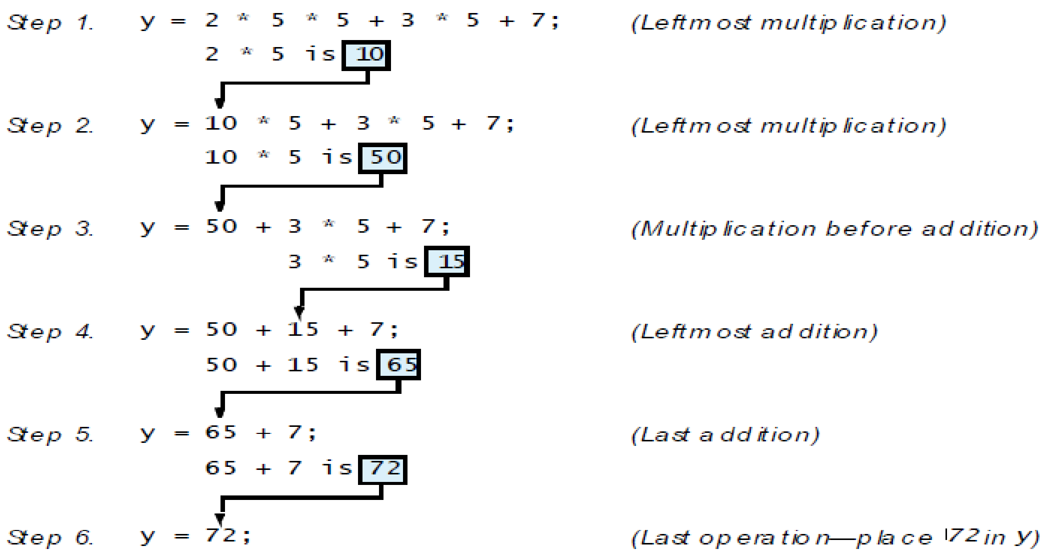


( )	for grouping the variables
_	(unary for negative number)
* /	(multiplication and division)
+ -	(addition and subtraction)

Example: X+y*X-Z, where X=5, Y=6, and Z=8.

5 + (6*5)-8 → (5+30)-8 → 35-8 → 27

Example:



```
Step 1.   y = 2 * 5 * 5 + 3 * 5 + 7;        (Leftmost multiplication)
              2 * 5 is 10
Step 2.   y = 10 * 5 + 3 * 5 + 7;           (Leftmost multiplication)
              10 * 5 is 50
Step 3.   y = 50 + 3 * 5 + 7;               (Multiplication before addition)
                   3 * 5 is 15
Step 4.   y = 50 + 15 + 7;                  (Leftmost addition)
              50 + 15 is 65
Step 5.   y = 65 + 7;                       (Last addition)
              65 + 7 is 72
Step 6.   y = 72;                           (Last operation—place 72 in y)
```

Example :

Write the following equation as a C++ expression:

$$f = \frac{a + b + c + d + e}{10}$$

Solution:

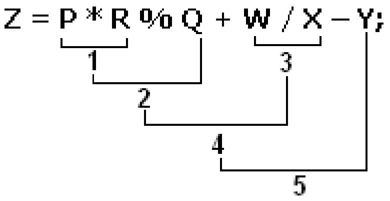
f = (a + b + c + d + e) / 10;

Note: the parentheses here are required because division has higher precedence than addition.

Example :

State the order of evaluation for the following expression:

Z = P * R % Q + W / X – Y;



Solution:

1. *
2. %
3. /
4. +
5. –

Example:

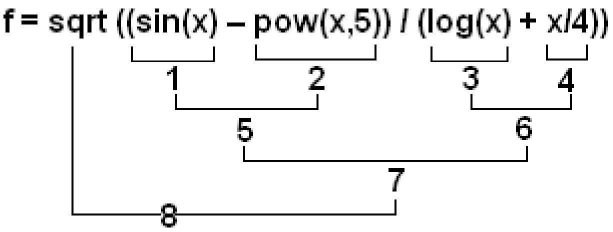
Write the following equation as a C++ expression and state the order of evaluation of the binary operators:

$$f = \sqrt{\frac{\sin(x) - x^5}{\ln(x) + \frac{x}{4}}}$$

Solution:

f = sqrt ((sin(x) – pow(x,5)) / (log(x) + x/4))

Order of evaluation:



```
#include<iostream>

#include<cmath>

Using namespace std;

Int main()

{

Int x;float f;

Cout<<"input x value"<<endl;

Cin>>x;

F=sqrt((sin(x)+pow(x,5))/(log(x)+x/4));

Cout<<"f="<<f<<endl;

}
```

Write C++ program to perform the above equation:

Z= P * R % Q + W / X - Y

```
#include<iostream>
Using namespace std;
Int main( )
{
int Z, P, R, Q, W, X, Y;
cout << "enter P:";
cin >> P;
cout << "enter R:";
 cin >> R;
cout << "enter Q:";
cin >> Q;
cout << "enter W:";
 cin >> W;
cout << "enter X:";
cin >> X;
cout << "enter Y:";
 cin >> Y;
Z= P * R % Q + W / X - Y;
cout << "the result="<< Z;

return 0;

}
```

Assignment Operators:

The operatonal assignment operator has the

form:

1-Variable = variable operator expression;

Ex: x=x+5; y=y*10;
The operational assignment operator can be written in the following
form:
2-Variable operator = expression
Ex: x+=5; y*=10;

S+=r; s=s+r;

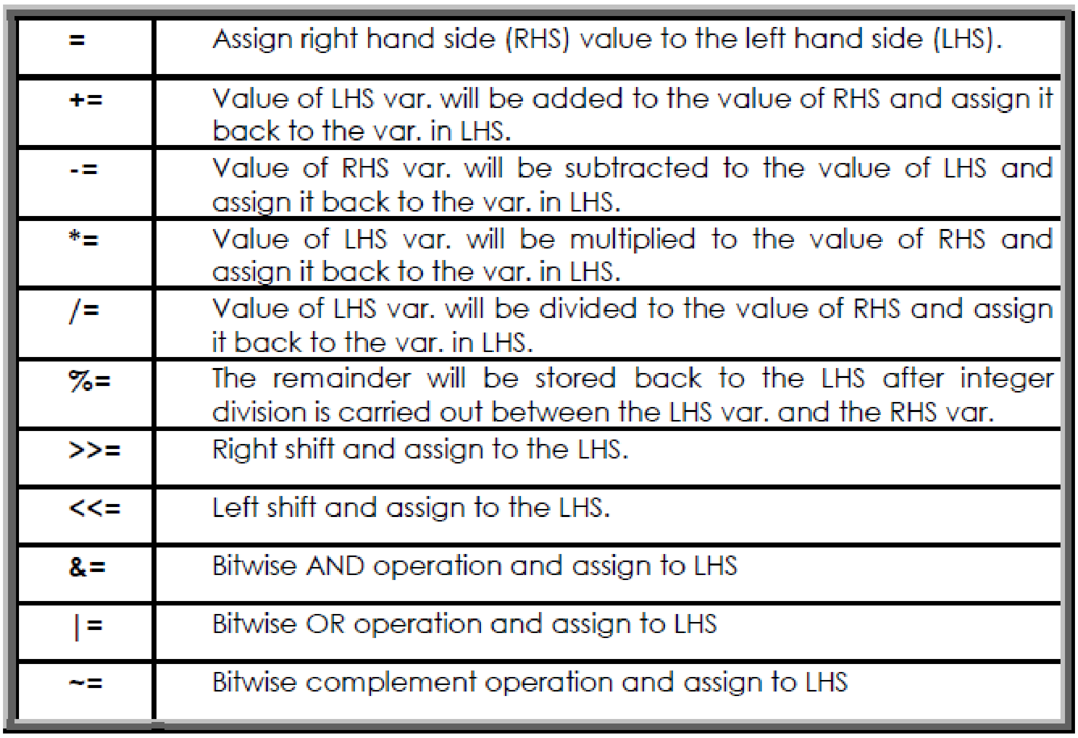
s-=r;   s=s-r;

s*=r; s=s*r;

s/=r; s=s/r;

s%=r; s=s%r;

It is used to assign back to a variable, a modified value of the present holding:



=	Assign right hand side (RHS) value to the left hand side (LHS).
+=	Value of LHS var. will be added to the value of RHS and assign it back to the var. in LHS.
-=	Value of RHS var. will be subtracted to the value of LHS and assign it back to the var. in LHS.
*=	Value of LHS var. will be multiplied to the value of RHS and assign it back to the var. in LHS.
/=	Value of LHS var. will be divided to the value of RHS and assign it back to the var. in LHS.
%=	The remainder will be stored back to the LHS after integer division is carried out between the LHS var. and the RHS var.
>>=	Right shift and assign to the LHS.
<<=	Left shift and assign to the LHS.
&=	Bitwise AND operation and assign to LHS
|=	Bitwise OR operation and assign to LHS
~=	Bitwise complement operation and assign to LHS

Exercise:

Rewrite the equivalent statements for the following examples, and find it results. Assume: X=2 , Y=3 , Z=4 , V=12 , C=8.

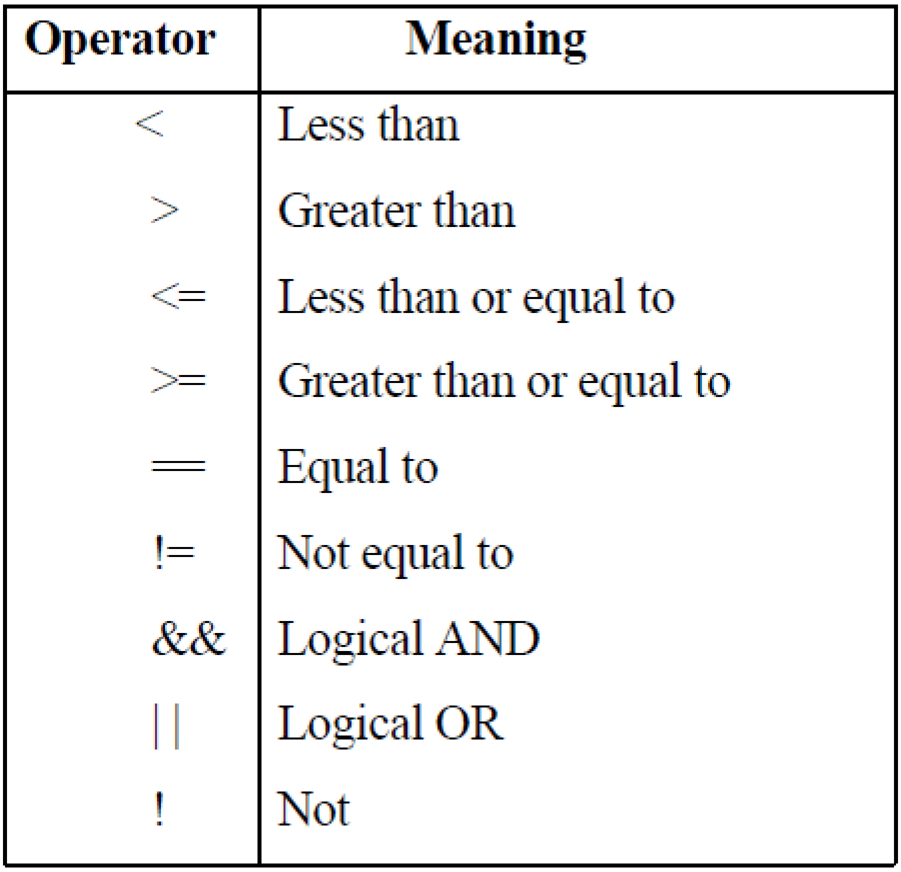
Example	Equivalent Statement	Result
X += 5	X = X + 5	X ← 7
Y -= 8	Y = Y - 8	Y ← -5
Z *= 5	Z = Z * 5	Z ←
V /= 4		V ←
C %= 3		C ←

```
// compound assignment operators

#include <iostream>
using namespace std;
int main ()
{
int a, b=3;
a = b;
a+=2; // equivalent to a=a+2
cout << a;
return 0;
}
```

```
// assignment operator
#include <iostream>
using namespace std;
int main ()
{
int a, b; // a:?, b:?
a = 10; // a:10, b:?
b = 4; // a:10, b:4
a = b; // a:4, b:4
b = 7; // a:4, b:7
cout << "a:";
cout << a;
cout << " b:";
cout << b;
return 0;
}
```

Comparison and Logical Operators

a)Relational operators:



Operator	Meaning
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to
&&	Logical AND
||	Logical OR
!	Not

Ex: 3 > 4 → false, 6 <=2 →false, 10>-32 → true, (23*7)>=(-67+89) → true

(b) Equality operators:

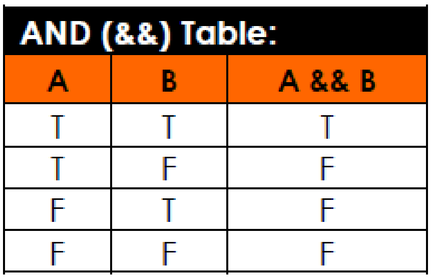
== equal to , **!=** not equal

Ex: a=4, b=6, c=8.
A==b→false, (a*b)!=c→true,‘s’==’y’ →false.

(c) Logical operators:

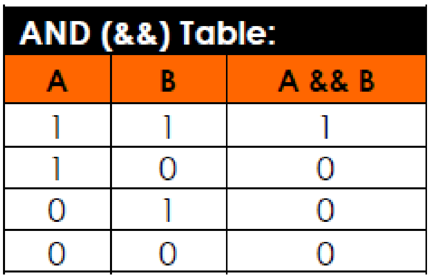
The logical expression is constructed from relational expressions by the use of the logical operators

not(!), and(&&), or(||).



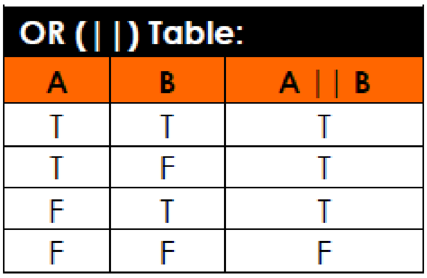
AND (&&) Table:

A	B	A && B
T	T	T
T	F	F
F	T	F
F	F	F



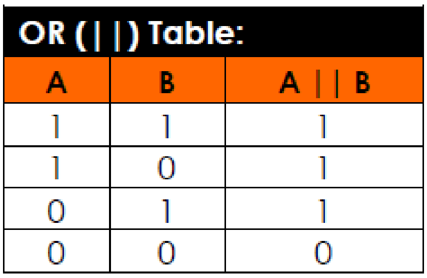
AND (&&) Table:

A	B	A && B
1	1	1
1	0	0
0	1	0
0	0	0



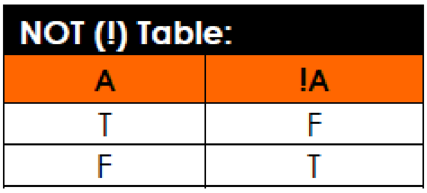
OR (| |) Table:

A	B	A | | B
T	T	T
T	F	T
F	T	T
F	F	F



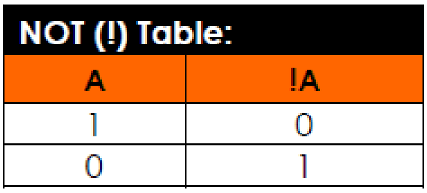
OR (| |) Table:

A	B	A | | B
1	1	1
1	0	1
0	1	1
0	0	0



NOT (!) Table:

A	!A
T	F
F	T



NOT (!) Table:

A	!A
1	0
0	1

Example:

a=4, b=5, c=6

(a<b)&&(b<c)	(a<b) | | (b>c)	!(a<b) | | (c>b)	(a<b) | | (b>c)&&(a>b) | | (a>c)
T && T	T | | T	!(T) | | T	T | | F && F | | F
T	T	F | | T	T | | F | | F
		T	T | | F
			T

Example :

Assume: X=0, Y=1, Z=1. Find the following expression:
M = ++X || ++Y && ++Z
= 1 || (2 && 2)
= T || (T && T)
= T || T
= T
= 1

Example:

(1)
a =4, b= 5, and c=6
(a < b) && (b < c)
(4 < 5) && (5 < 6)
true && true
true

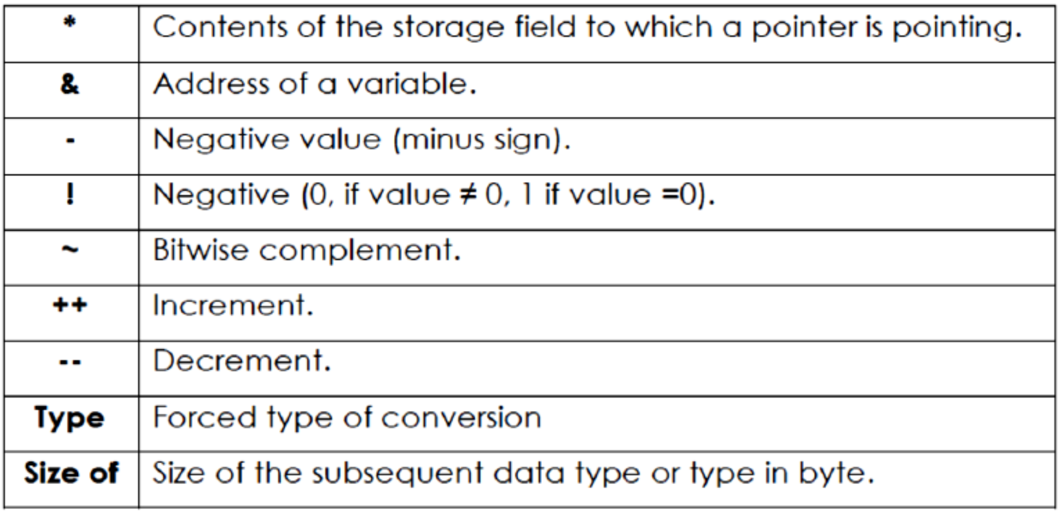
Example:

(2)
a = 4 , b = 5, and c = 6
(a < b) | | (b > c)
(4 < 5) | | (5 > 6)
true | | false
true

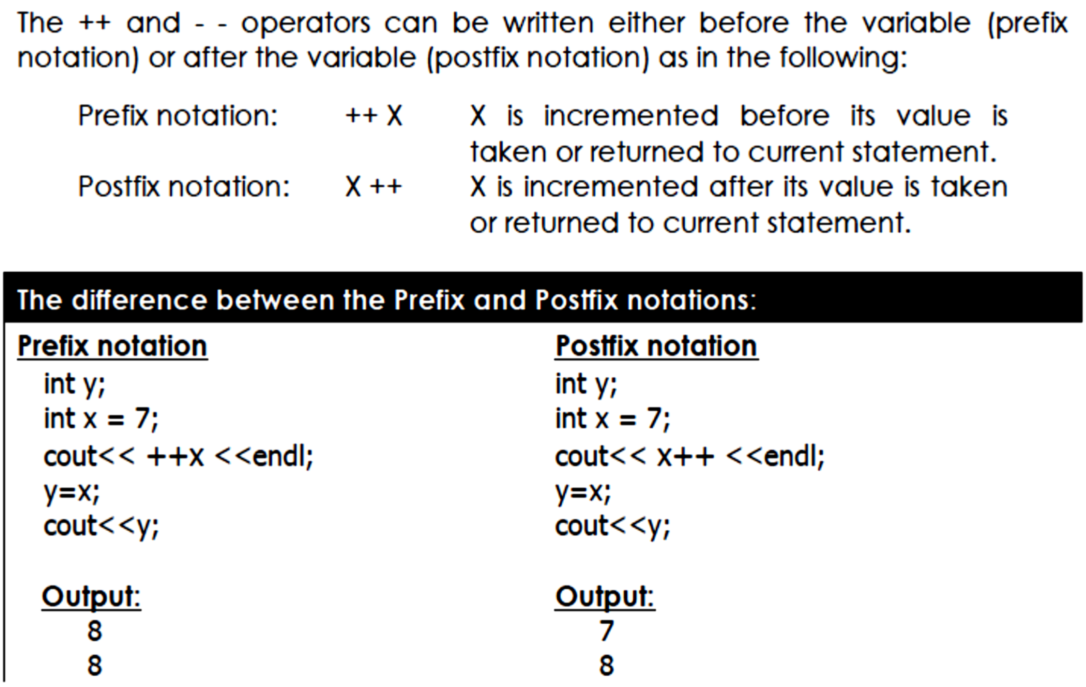
Example:

(3)
a = 4 , b = 5 , and c = 6
! (a < b) | | (c > b)
! (4 < 5) | | (6 > 5)
! (true) | | true
false | | true
true
This expression evaluates to true.

Special operators:



*****	Contents of the storage field to which a pointer is pointing.
&	Address of a variable.
-	Negative value (minus sign).
!	Negative (0, if value ≠ 0, 1 if value =0).
~	Bitwise complement.
++	Increment.
--	Decrement.
Type	Forced type of conversion
Size of	Size of the subsequent data type or type in byte.



The ++ and - - operators can be written either before the variable (prefix notation) or after the variable (postfix notation) as in the following:

Prefix notation:	++ X	X is incremented before its value is taken or returned to current statement.
Postfix notation:	X ++	X is incremented after its value is taken or returned to current statement.

The difference between the Prefix and Postfix notations:

Prefix notation

```
int y;
int x = 7;
cout<< ++x <<endl;
y=x;
cout<<y;
```

Output:

```
8
8
```

Postfix notation

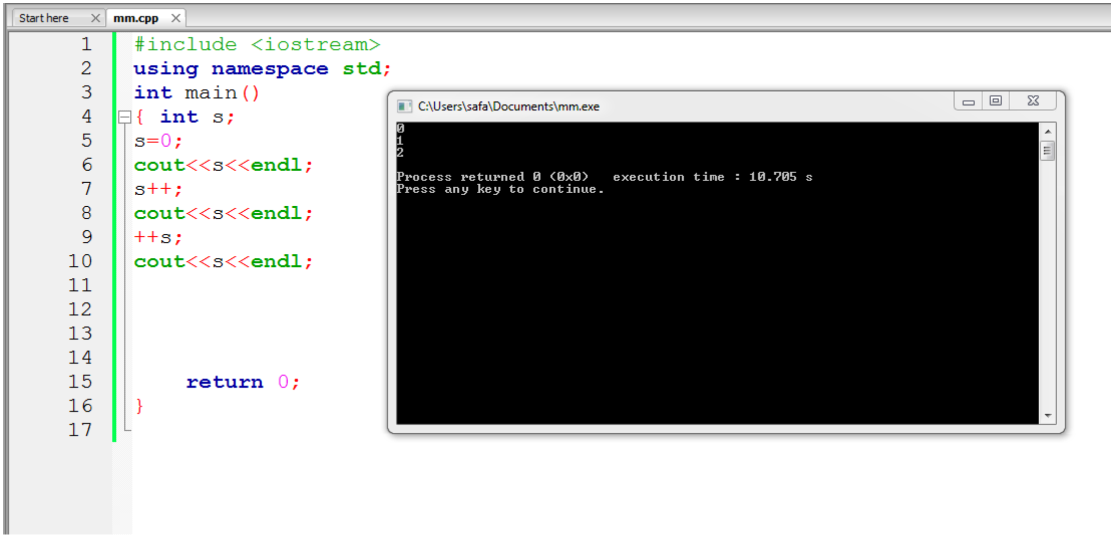
```
int y;
int x = 7;
cout<< x++ <<endl;
y=x;
cout<<y;
```

Output:

```
7
8
```

display the output of the following program:

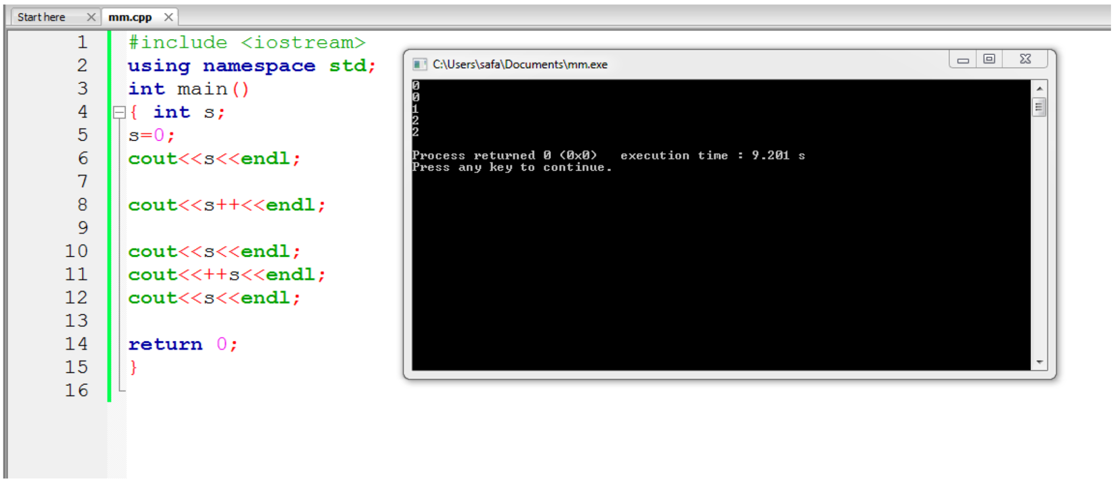
Example:



```
#include <iostream>
using namespace std;
int main()
{ int s;
s=0;
cout<<s<<endl;
s++;
cout<<s<<endl;
++s;
cout<<s<<endl;




    return 0;
}
```

```
0
1
2

Process returned 0 (0x0)   execution time : 10.705 s
Press any key to continue.
```

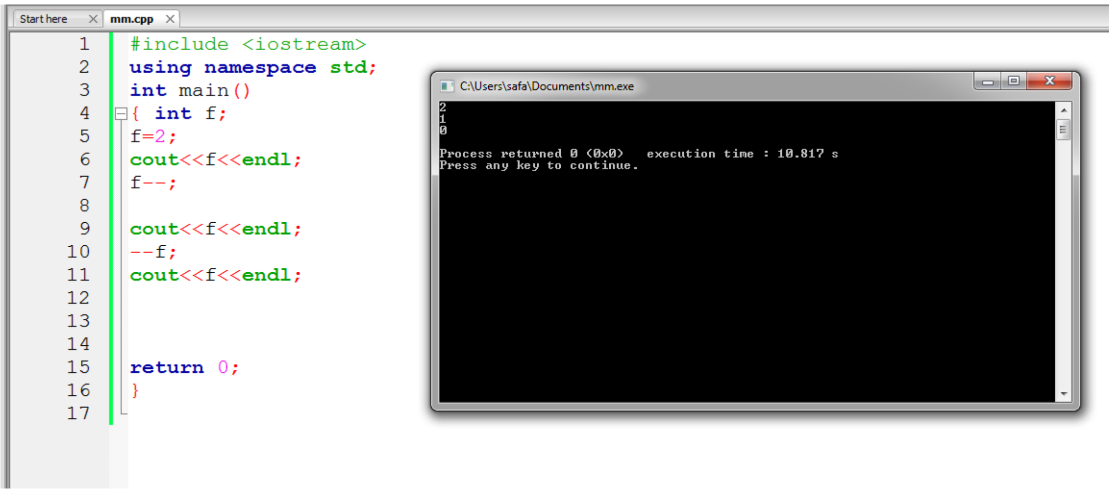
Example:



```
#include <iostream>
using namespace std;
int main()
{ int s;
s=0;
cout<<s<<endl;

cout<<s++<<endl;

cout<<s<<endl;
cout<<++s<<endl;
cout<<s<<endl;

return 0;
}
```

```
0
0
1
2
2

Process returned 0 (0x0)   execution time : 9.201 s
Press any key to continue.
```

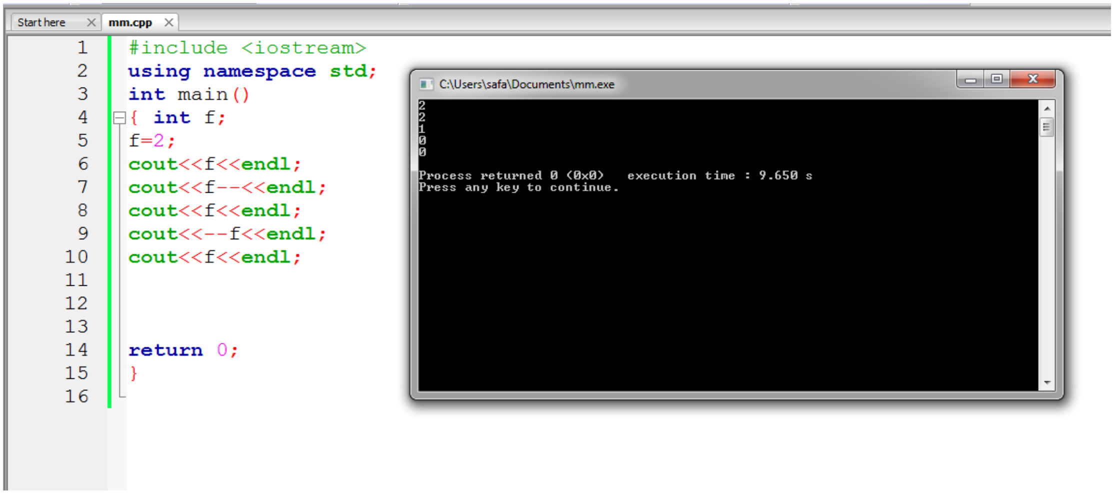
Example:



```
#include <iostream>
using namespace std;
int main()
{ int f;
f=2;
cout<<f<<endl;
f--;

cout<<f<<endl;
--f;
cout<<f<<endl;



return 0;
}
```

```
2
1
0

Process returned 0 (0x0)   execution time : 10.817 s
Press any key to continue.
```

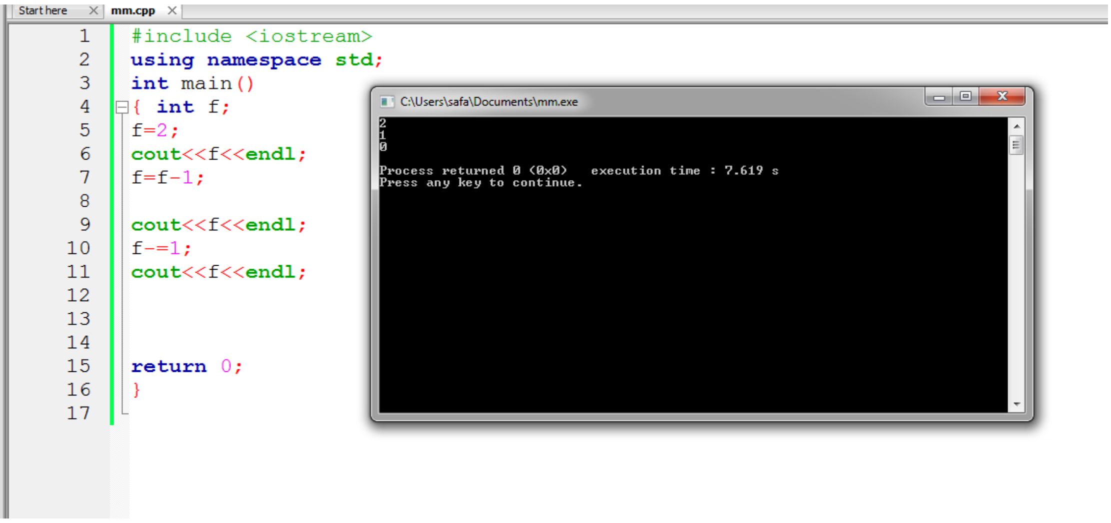
Example:



```cpp
#include <iostream>
using namespace std;
int main()
{ int f;
f=2;
cout<<f<<endl;
cout<<f--<<endl;
cout<<f<<endl;
cout<<--f<<endl;
cout<<f<<endl;



return 0;
}
```

```
2
2
1
0
0

Process returned 0 (0x0)   execution time : 9.650 s
Press any key to continue.
```

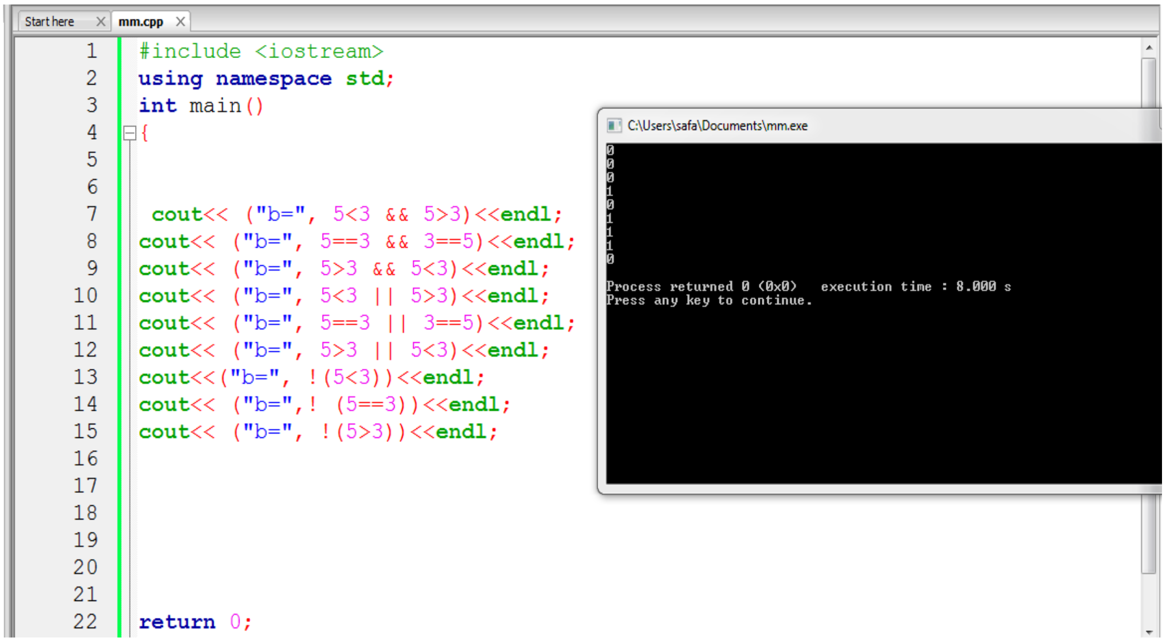
Example:



```cpp
#include <iostream>
using namespace std;
int main()
{ int f;
f=2;
cout<<f<<endl;
f=f-1;

cout<<f<<endl;
f-=1;
cout<<f<<endl;



return 0;
}
```

```
2
1
0

Process returned 0 (0x0)   execution time : 7.619 s
Press any key to continue.
```

Example:



```cpp
#include <iostream>
using namespace std;
int main()
{


  cout<< ("b=", 5<3 && 5>3)<<endl;
cout<< ("b=", 5==3 && 3==5)<<endl;
cout<< ("b=", 5>3 && 5<3)<<endl;
cout<< ("b=", 5<3 || 5>3)<<endl;
cout<< ("b=", 5==3 || 3==5)<<endl;
cout<< ("b=", 5>3 || 5<3)<<endl;
cout<<("b=", !(5<3))<<endl;
cout<< ("b=",! (5==3))<<endl;
cout<< ("b=", !(5>3))<<endl;






return 0;
```

```
0
0
0
1
0
1
1
1
0

Process returned 0 (0x0)   execution time : 8.000 s
Press any key to continue.
```

Example:

(1)Write a program to find avarage of three students

```
#include<iostream>
using namespace std;
int main()
{
float d1,d2,d3,sum,avarage;
cout<<"enter tht three degree";
cin>>d1>>d2>>d3;
sum=d1+d2+d3;
avarage=sum/3;
cout<<"the avarages of three degree"<<avarage;
return 0;
}
```

(2)Write a c++ program to convert temperature in Fahrenheit to centigrade where c=(f-32)*5/9

```
#include<iostream>
Using namespace std;
Int main()
{
Int f,c;
Cin>>f;
C=(f-32)*5/9.0;
Cout<<"c   \n"<<c;
Return 0;
}
```

(3)The length ,width of rectangle and radius of a circle are input through the keyboard.write aprogram to calculate the area&perimeter of the rectangle ,and the area&circumference of the circle.

```
#include<iostream>
using namespace std;
int main()
{
   float l,w,r,a1,a2,m1,m2;

   cout<<"enter lenth"<<endl;
   cout<<"enter width"<<endl;
   cout<<"enter radius"<<endl;
   cin>>l>>w>>r;
   a1=l*w;
  m1=(l+w)*2;
   a2=r*r*3.14;
   m2=2*3.14*r;
   cout<<"a1="<<a1<<endl;
    cout<<"m1="<<m1<<endl;
     cout<<"a2="<<a2<<endl;
cout<<"m2="<<m2<<endl;
return 0;
}
```

(4)Two numbers are input through the keyboard into two location c and d.write a program to interchange the contents of c and d.

```
#include<iostream>
using namespace std;
int main()
{
   int a,c,d;
   cout<<"enter the value of c"<<endl;
    cout<<"enter the value of d"<<endl;
   cin>>c>>d;
   a=c;
   c=d;
   d=a;
   cout<<"the result="<<c<<endl;
    cout<<"the result="<<d;
return 0;
}
```

(5)Write in c++ program to convert the time in (h,m,s)to second.

```
#include<iostream>
using namespace std;
int main()
{int h,m,s,z;
cout<<"enter the time in(h,m,s)";
cin>>h>>m>>s;
z=h*3600+m*60+s;
cout<<"the time in seconds"<<z;
return 0;
}
```

(6)Write a c++ program to print first digit from integer number.

```
#include<iostream>
using namespace std;
int main()
{
  int num,fd;
  cout<<"enter the number";
  cin>>num;
  fd=num%10;
  cout<<"the first digit="<<fd;
return 0;
}
```

(7)Write ac++ program to print the second digit of numbers.

```
#include<iostream>
using namespace std;
int main()
{
  int num,fd;
  cout<<"enter the number";
  cin>>num;
  num=num%10;
  fd=num/10;
 cout<<"the second digit is="<<fd;
return 0;
}
```

(8)Write a c++ program to convert any lower case char into upper case char.

```
#include<iostream>
using namespace std;
int main()
{
  int lc,uc;
  cout<<"enter the number";
  cin>>lc;
  uc=lc-32;
  cout<<"upper case="<<uc;
   return 0;
    }
```

Exercise:

Write the following equation as a C++ expression and state the order of evaluation of the binary operators:

$$Z = \sqrt{\frac{x^2 y - 3 \sin(x)}{\tan x^3 + x^3 / y}}$$

Solution: ?

Example:

a = 4 , b = 5, and c = 6

(1)
(a < b) | | (b > c) && (a > b) | | (a > c)

(2)
((a < b) | | (b > c)) && ((a > b) | | (a > c))

(3)
! ((a < b) | | (b > c))

What is the Output:

(1)

```
#include<iostream>

Using namespace std;

Int main()

{

Int total=0;

Int count=10;

Cout<<total;

Total=total+count;

Cout<<total;

Total+=count;

Cout<<total;

Return 0;

}
```

(2)

Find the value of x for each step.

```
#include<iostream>

Using namespace std;

Int main()

{ int x,y,z;

X=y=z=0;

X=++y    +  ++z;

X=y++ + z++;

X=++y  +  z++;

X=y--  + --z;

}
```

(3)

Find the o/p of the following program

```
{int x,y,z;
X=4;y=5;
Z=x++ *y;
Cout<<x<<z;
Z=++x*y;
Cout<<x<<z;
}
```

Solve the program:

1)Temperature of a city in Fahrenheit degrees is input through the keyboard.write aprogram to convert this temperature into centigrade degree.

2) If a five-digit number is input through the keyboard,write a program to reverse the number.

3) Write in c++ program to convert the time in second to (h,m,s).

4) Write a program in c ++ to insert a number contain 3 digit and then print the inverse.

5) If the total selling price of 15 items and the total profit earned on them is input through the keyboard.write a program to find the cost price of one item.

AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

1st STAGE STUDENTS

Assistant lecturer

Safa Hussain

Selection(conditional)statement

Conditional expressions are mainly used for decision making. C++ provides multiple selection structures:

1)if statement

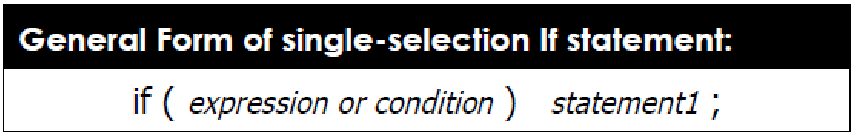
2)if…….else statements

3)Nested if statements

4)switch statement

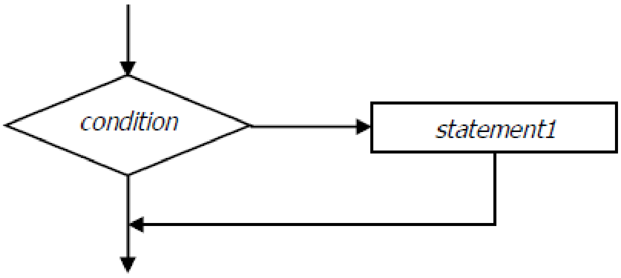
If statement

The IF statement is used to express conditional expression. If the given condition is true then it will execute the statements; otherwise it will execute the optional statements.

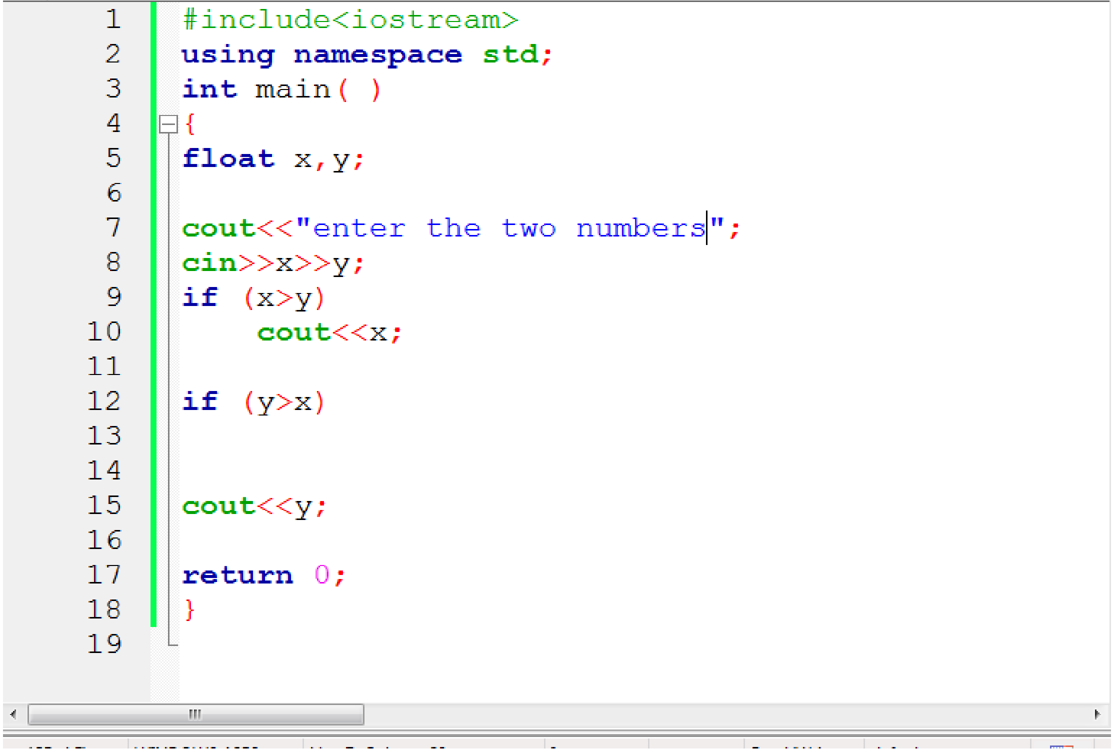


General Form of single-selection If statement:

if (*expression or condition*) *statement1* ;



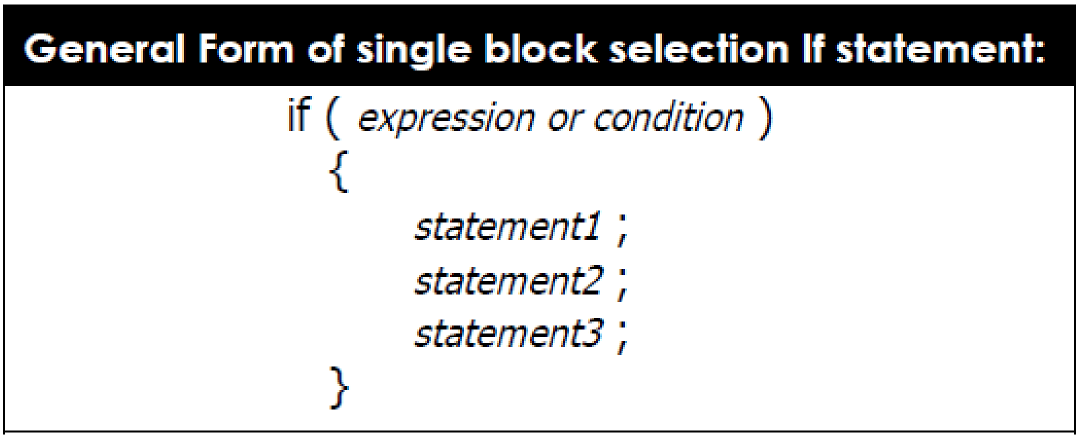
Example: Write a C++ program to read any two numbers and print the largest value of it:



```
#include<iostream>
using namespace std;
int main( )
{
float x,y;

cout<<"enter the two numbers";
cin>>x>>y;
if (x>y)
    cout<<x;

if (y>x)


cout<<y;

return 0;
}
```

The Single Block If Statement Structure :

The block IF statement are enclosed in ({) and (}) to group declaration and statements into a compound statement or a block. These blocks are always considered as a single statement. The structure is:



General Form of single block selection If statement:

```
if ( expression or condition )
  {
        statement1 ;
        statement2 ;
        statement3 ;
  }
```

Example: Write a C++ program to read a number and check if it’s positive, if it’s so print it, decrement it by 2:

```
#include<iostream>

using namespace std;

int main( )

{

int num;


cin>> num;

if(num>=0)

{

 cout<<"positive"<<num<<endl;

num=num-2;

cout<<num;

}

return 0;

}
```

Example:Write a program that adds five (5) to the average student rate(45-50)

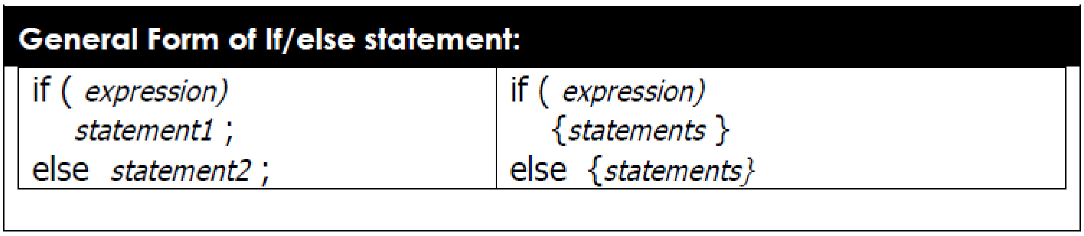
```
#include<iostream>

using namespace std;

int main( )

{

float av;

cin>> av;

if(av>=45&&av<50)

{

av=av+5;

  cout<<"success"<<av<<endl;

}return 0;

}
```

Assistant lecturer Safa Hussain

Example:write a c++ program to input integer number then print if it's consist of 2 digit or not.

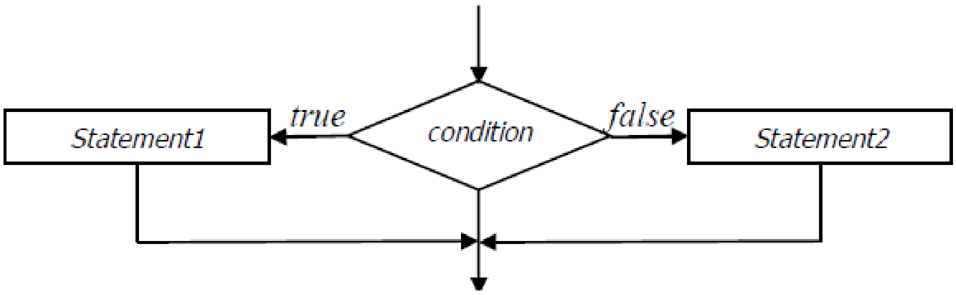
```
#include<iostream>

using namespace std;

int main( )

{

int x;

cin>>x;

if(x>9&&x<100)

cout<<"yes"<<x;


if(x<10||x>99)

   cout<<"no"<<x;


return 0;

}
```

If……else statement



General Form of If/else statement:	

if (*expression)*
statement1 ;
else *statement2* ;
	if (*expression)*
	{*statements* }
	else {*statements}*



Example: Write a program that tests if the number is positive or negative

```
#include<iostream>
using namespace std;
int main( )
{
int a;
cin >> a;
if(a>=0)
cout<<"positive";
else
cout << "negative";

return 0;
}
```

Example: Write a C++ program to read a student degree, and check if it’s degree greater than or equal to 50, then print pass, otherwise print fail:

```
#include<iostream>
using namespace std;
int main( )
{
int degree;
cin >> degree;
if (degree >= 50 )
cout << "pass";
else
cout << "fail";
return 0;
}
```

Example:write a program that tests if the number is even or odd number and adds it 4 to even number only.

```
#include<iostream>

using namespace std;

int main( )

{

int x;

cin>>x;

if(x%2==0)

{

x=x+4;

   cout<<"even"<<endl<<x;

}

   else

    cout<<"odd"<<endl<<x;

return 0;

}
```

Else if Statements:

```
if ( expression or condition 1 )

statement1 ;
else
 if ( expression or condition 2 )
statement2 ;
else
if ( expression or condition 3 )
statement3 ;
:
.
.
Else
 if ( expression or condition n )
 statement-n ;
else
 statement-e ;
```

Assistant lecturer Safa Hussain

Example:write a c++ program to input one character from the keyboard then print type of character.

```
#include<iostream>
using namespace std;
int main( )
{
char x;
cin>>x;
if(x>='A'&&x<='Z')
    cout<<"upper case"<<endl<<x;
else
if(x>='a'&&x<='z')
 cout<<"lower case"<<endl<<x;
else
if(x>='0'&&x<='9')
cout<<"digital"<<endl<<x;
else
cout<<"special"<<endl<<x;
return 0;
}
```

Example: Write a c++ program to input the average for one student ,then print grade of student.

```
#include<iostream>
using namespace std;
int main( )
{
float av;
cin>>av;
if(av>=50&&av<=59)
    cout<<"your grade is pass"<<endl;
else
if(av>=60&&av<=69)
    cout<<"your grade is medium"<<endl;
else
    if(av>=70&&av<=79)
    cout<<"your grade is good"<<endl;
else
    if(av>=80&&av<=89)
    cout<<"your grade is verygood"<<endl;
else
    if(av>=90&&av<=100)
    cout<<"your grade is excelent"<<endl;
else
cout<<"your grade is fail"<<endl;

return 0;
}
```

Example:write a c++ program to input one character and print the next character according to the following table

```
#include<iostream>
using namespace std;
int main( )
{
char x;
cin>>x;
if(x>='A'&&x<='Y'||x>='a'&&x<='y'||x>'0'&&x<'8')
   x++;
else
   if(x=='Z')
   x='A';
   else
   if(x=='z')
   x='a';
else
   if(x=='9')
   x='0';
else
   x='*';
cout<<x;

return 0;
}
```

Example: Write C++ program to compute the value ofz according to the following equations:

$$Z = \begin{cases} x + 5 & : x < 0 \\ \cos(x) + 4 & : x = 0 \\ \sqrt{x} & : x > 0 \end{cases}$$

```
#include<iostream>
#include<math.h>
using namespace std;
int main( )
{
float x,z;
cout << "Enter X value \n";
cin>>x;
if(x<0)
   z=x+5;
else
if(x==0)
z=sin(x)+4;

else
z=sqrt(x);
cout << "z is " <<z;
return 0;
}
```

The Switch Selection Statement

```
switch ( selector )
{
case label1 :
statement1 ;
 break;
case label2 :
 statement2 ;
 break;
case label3 :
statement3 ;
 break;
:
case label-n :
statement-n ;
break;
default :
statement-e ;
break;
}
```

Example: Write C++ program to read integer number, and print the name of the day in a week:

```
#include<iostream>
using namespace std;
int main( )
{
int day;
cout << "Enter the number of the day \n";
cin >> day;
switch (day)
{
case 1:
cout << "Sunday";
 break;
case 2:
cout << "Monday";
    break;
case 3:
cout << "Tuesday";
 break;
case 4:
     cout << "Wednesday";
     break;
case 5:
     cout << "Thursday";
      break;
case 6:
     cout << "Friday";
      break;
case 7:
     cout << "Saturday";
      break;
default:
    cout << "Invalid day number";
     break;
}
return 0;
}
```

Example: Write C++ program to read two integer numbers, and read the operation to perform on these numbers:

```
#include<iostream>
using namespace std;
int main( )
{
int a, b;
char x;
cout << "Enter two numbers \n";
cin >> a >> b;
cout << "+ for addition \n";
cout << "- for subtraction \n";
cout << "* for multiplication \n";
cout << "/ for division \n";
cout << "enter your choice \n";
cin >> x;
switch ( x )
{
case '+':
cout << a + b;
break;
case '-':
cout << a - b;
break;
case '*':
cout << a * b;
break;
case '/':
cout << a / b;
break;
default:
break;

}
return 0;
}
```

Example: Write a program to see if the entered character is a vowel or not.

```
#include<iostream>
using namespace std;
int main( )
{
char x;
cin>>x;
switch(x)
{
case'a':
    cout<<"vowel";
    break;
    case'i':
    cout<<"vowel";
     break;
    case'e':
    cout<<"vowel";
     break;
    case'o':
    cout<<"vowel";
     break;
    case'u':
    cout<<"vowel";
    break;
    default:
        cout<<"invaild vowel";
}
return 0;
}
```

Homework:

1- Write a c++ program to find a large number between three integer number?

2- Write a c++ program that performs the arithmetic operation (+,-,*,/)

Determined by a user input.

3- Input integer number which consist of 3 digits and find out the maxiumum digit.

4- Write a C++ program to read a number, and print the day of the

week:

5- Write a c++ program to read the month number then print the name of the month(switch-case)

6- Write a c++ program to input the average for one student ,then print grade of student(switch-case)

AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

1st STAGE STUDENTS

Assistant lecturer

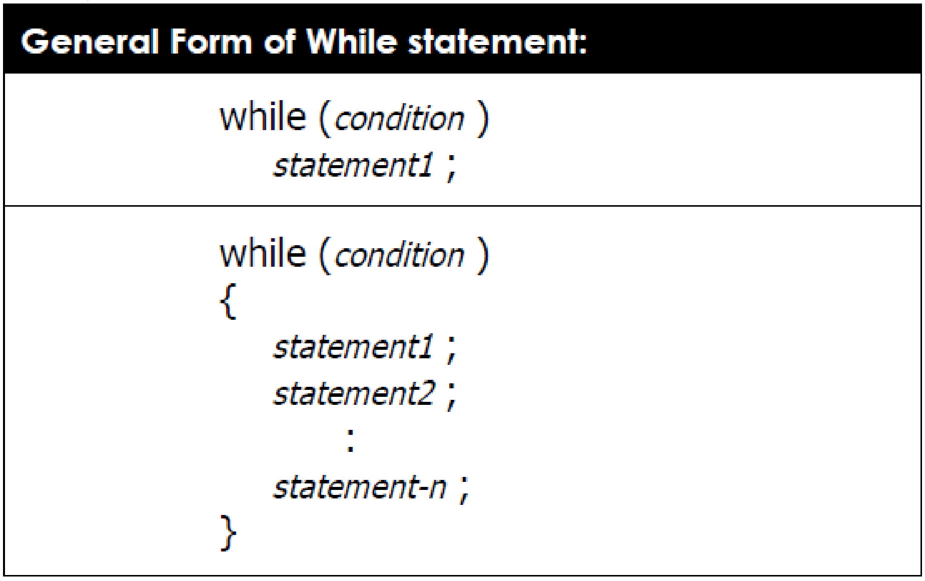
Safa Hussain

Loop Statements:

The loop statements are essential to construct systematic block styled programming. C++ provides three iteration structures:

1- **while**
2- **do/while**
3- **for**.

1)While



General Form of While statement:

```
while (condition )
    statement1 ;
```

```
while (condition )
{
    statement1 ;
    statement2 ;
        :
    statement-n ;
}
```

Example:

```
i = 0;
while ( i < 10 )
{
cout << i;
i ++;
}
```

Output:

```
0 1 2 3 4 5 6 7 8 9
```

Example:

```
i = 1;
while ( i < 10 )
{
cout << i;
i =i+ 2;
}
```

Output: ***odd numbers only***

```
1 3 5 7 9
```

Example:Write C++ program to find the summation of the following series:

sum = 1 + 3 + 5 + 7 + … + 99

***in other words: find the summation of the odd numbers, between 0 and 100)*.**

```
#include<iostream>
using namespace std;
int main( )
{
int i = 1;
int sum = 0;
while ( i <= 99 )
{
sum = sum + i;
i = i + 2;
}
cout << "sum is: " << sum << endl;
return 0;
}
```

Example:Write C++ program to find the cub of a number, while it is positive:

```
#include<iostream>
using namespace std;
int main( )
{
int num, cubenum;
cout << "Enter positive number \n";
cin >> num;
while ( num > 0 )
{
cubenum = num*num*num;
cout << "cube number is :" << cubenum << endl;
}
return 0;
}
```

Example:Write C++ program to find the summation of the following series:

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2$$

```
#include<iostream>
using namespace std;
int main( )
{
int i = 1, n ,sum = 0;
cout << "enter positive number";
cin >> n;
while ( i <= n )
{
sum =sum + i * i ;
i++;
}
cout << "sum is: " << sum << endl;
return 0;
}
```

Example:Write C++ program to print the following using while:

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

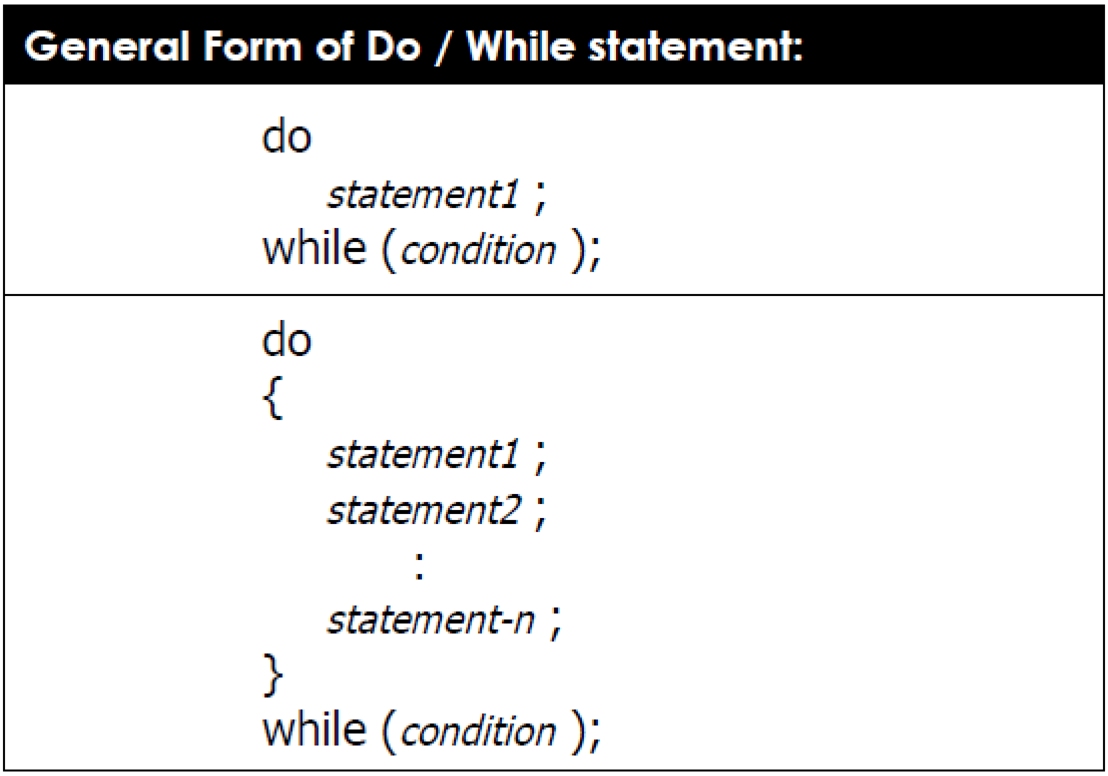
```
#include <iostream>
using namespace std;
int main()
{
int i=1;
while (i<=4)
{
int j=1;
while(j<=4)
{
cout<<"* ";
j++;
}
cout<<"\n";
i++;
}
return 0;}
```

Example:write c++ program using while to print the following list number:

5 15 25 35 45 55 65 75 85 95

```
#include<iotream>
Using namespace std;
Int main()
{
Int i=5;
While(i<=95)
{
Cout<<i<<endl;
i=i+10;
}
return 0;
}
```

Do / While Statement:



General Form of Do / While statement:

```
do
    statement1 ;
while (condition );
```

```
do
{
    statement1 ;
    statement2 ;
         :
    statement-n ;
}
while (condition );
```

Example

```
i = 0;
do
{
cout << i;
i ++;
}
while ( i < 10 )
Output:      0 1 2 3 4 5 6 7 8 9
```

Assistant lecturer Safa Hussain

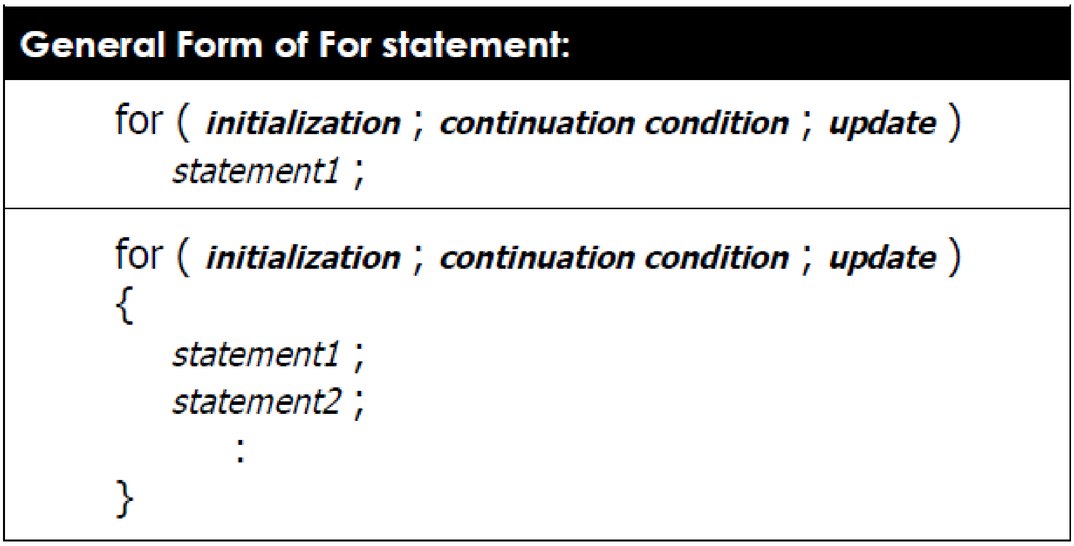
Example:Write C++ program to valid input checking, that accept the numbers between 50 ... 70 only:

```
#include<iostream>
using namespace std;
int main( )
{
int accept;
int x,low=50,high=70;
do
{
cout <<"enter number: ";
cin >> x;
if (x>=low&&x<=high )
cout<<"accept";
else
cout<<"not accept";
}
while(accept==1);
return 0;
}
```

Example:Write C++ program to find the factorial of n:
n! = n * n-1 * n-2 * n-3 * … * 2 * 1;

```
#include<iostream>
using namespace std;
int  main( )
{
int n, f = 1;
cout << "enter positive number: ";
cin >> n;
do
{
f = f * n;
n --;
}
while ( n > 1 );
cout << "factorial is: " << f;
return 0;
}
```

For



General Form of For statement:

```
for ( initialization ; continuation condition ; update )
    statement1 ;
```

```
for ( initialization ; continuation condition ; update )
{
    statement1 ;
    statement2 ;
        :
}
```

Example 1: for (i = 0; i < 10; i ++)
cout << i;

Output:
0 1 2 3 4 5 6 7 8 9

Example 2: for (i = 0; i < 10; i += 2)
cout << i;

Output: ***even numbers only***
0 2 4 6 8

Example 3: for (i = 1; i < 10; i += 2)
cout << i;

Output: ***odd numbers only***
1 3 5 7 9

Example:Write C++ program to add the numbers between 1 and 100:

```
#include<iostream>
using namespace std;
int main( )
{
int sum = 0;
for ( int i = 1; i <= 100; i ++ )
sum = sum + i;
cout << "sum is: " << sum;
return 0;
}
```

Example:Write C++ program to find the factorial of n (*using for statement*):

n! = n * n-1 * n-2 * n-3 * … * 2 * 1

```
#include<iostream>
Using namespace std;
Int main( )
{
int n, f = 1;
cout << “enter positive number: “;
cin >> n;
for ( int i = 1; i <= n; i ++ )
f = f * i;
cout << “factorial is: “ << f;
return 0;
}
```

Example:Write C++ program to the result of the following:

$$\sum_{i=1}^{20} a_i^2$$

```
#include<iostream>
Using namespace std;
int main( )
{
int sum = 0;
for ( int i = 1; i <= 20; i ++ )
sum = sum + ( i * i );
cout << “The sum is: “ << sum;
return 0;
}
```

Example:Write C++ program to read 10 integer numbers, and find the sum of positive number only:

```
#include<iostream>
Using namespace std;
int main( )
{
int num, sum = 0;
for ( int i = 1; i <= 10; i ++ )
{
cout << “enter your number: “;
cin >> num;
if ( num > 0 )
sum = sum + num;
}
cout << “The sum is: “ << sum;
return 0;
}
```

Homework:

1)Example:Write C++ program to find the summation of students marks,and it’s average, assume the student have 8 marks? Use(While)

2)Write a c++ program that prints the following shape:use(While)?

```
12345678910
123456789
12345678
1234567
123456
12345
1234
123
12
1
```

3) Write C++ program to find the summation of student’s marks, and it’s average, assume the student have 8 marks?use(d0-while)

4) Example:write a c++ program that computes the sum of integer numbers input by the user.the program should stop when the user enters zero.use(d0-while).

Assistant lecturer Safa Hussain

5) Write C++ program to print the following:(use for)

```
1   10
 2   9
 3   8
 4   7
 5   6
 6   5
```

6) Write C++ program to print the following series: 1, 2, 4, 8, 16, 32, 64 (use for)

AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

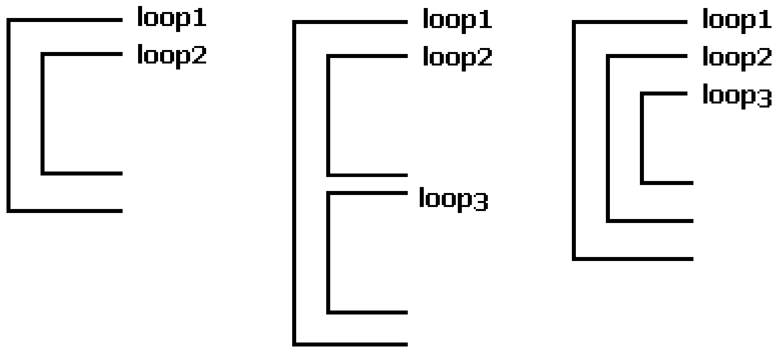
1st STAGE STUDENTS

Assistant lecturer

Safa Hussain

Nested Loops

We can put loops one inside another to solve a certain programming problems. Loops may be nested as follows:



Example:Write ac++ program to print the following figure:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

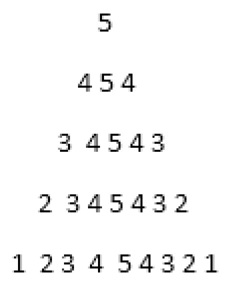
```
#include<iostream>
using namespace std;
int main()
{
   for(int i=0;i<=5;i++)
   {
      for(int j=1;j<=5;j++)
         if(i>=j)
         cout<<j;
      cout<<endl;
   }
}
```

Example: Write ac++ program to print the following figure:

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

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1;i<=5;i++)
{
   for(int j=1;j<=4;j++)
if(i<=j)
   cout<<" ";
for(int j=1;j<=5;j++)
   if(i>=j)
   cout<<"*";
   cout<<endl;
}
}
```

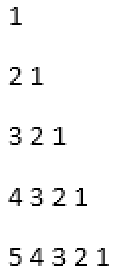
Example: Write ac++ program to print the following figure:



```
         5
       4 5 4
     3 4 5 4 3
   2 3 4 5 4 3 2
 1 2 3 4 5 4 3 2 1
```

```
#include<iostream>
using namespace std;
int main()
{
for(int i=5;i>=1;i--)
{
    for(int j=4;j>=1;j++)
   if(i>=j)
    cout<<"  ";
   for(int j=1;j<=5;j++)
    if(i<=j)
     cout<<j;
     for(int j=4;j>=1;j--)
      if(i<=j)
      cout<<j;
     cout<<endl;
        }
       }
```

Example: Write ac++ program to print the following figure:



```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1;i<6;i++)
{
for(int j=i;j>0;j--)
cout<<j;
 cout<<endl;
}
}
```

Jump statements:

(1)The break statement

Using break we can leave a loop even if the condition for its end is not fulfilled. It can be used to end an infinite loop, or to force it to end before its natural end.

Example:

```
#include <iostream>
using namespace std;
int main ()
{
int n;
for (n=10; n>0; n--)
{
cout << n << ", ";
if (n==3)
{
cout << "countdown aborted!";
break;
}
}
return 0;
}
```

Output: **10, 9, 8, 7, 6, 5, 4, 3, countdown aborted!**

(2)*The continue statement*

The continue statement causes the program to skip the rest of the loop in the current iteration as if the end of the statement block had been reached,causing it to jump to the start of the following iteration.

Example:

```
#include <iostream>
using namespace std;
int main ()
{
for (int n=10; n>0; n--) {
if (n==5) continue;
cout << n << ", ";
}
cout << "FIRE!\n";
return 0;
}
```

Output:

```
10, 9, 8, 7, 6, 4, 3, 2, 1, FIRE!
```

(3)*The goto statement*

goto allows to make an absolute jump to another point in the program. You should use this feature with caution since its execution causes an unconditional jump ignoring any type of nesting limitations.
The destination point is identified by a label, which is then used as an argument for the goto statement. A label is made of a valid identifier followed by a colon (:).

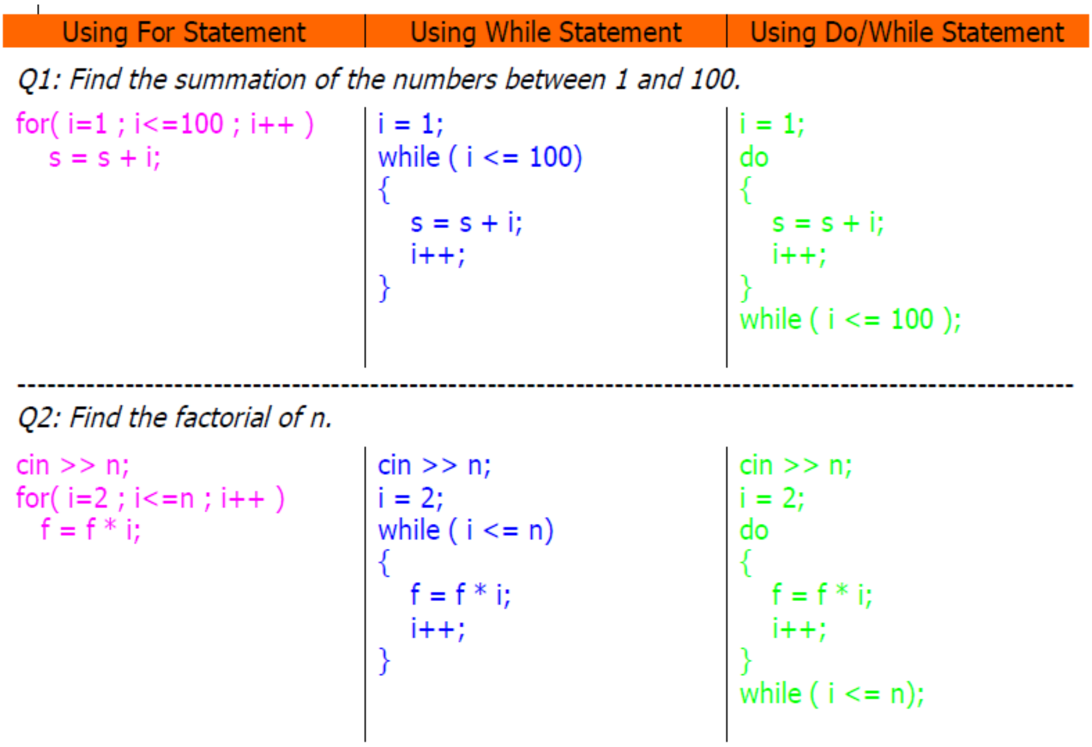
Example:

```
#include <iostream>
using namespace std;
int main ()
{
int n=10;
loop:
cout << n << ", ";
n--;
if (n>0) goto loop;
cout << "FIRE!\n";
return 0;
}
```

Output:

```
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, FIRE!
```

Note:



Using For Statement	Using While Statement	Using Do/While Statement
Q1: Find the summation of the numbers between 1 and 100.		

for(i=1 ; i<=100 ; i++)
s = s + i;
	i = 1;
	while (i <= 100)
	{
	s = s + i;
	i++;
	}
		i = 1;
		do
		{
		s = s + i;
		i++;
		}
		while (i <= 100);

Q2: Find the factorial of n.		

cin >> n;
for(i=2 ; i<=n ; i++)
f = f * i;
	cin >> n;
	i = 2;
	while (i <= n)
	{
	f = f * i;
	i++;
	}
		cin >> n;
		i = 2;
		do
		{
		f = f * i;
		i++;
		}
		while (i <= n);

Q3: To find the result of the following: $\sum_{i=1}^{20} a_i^2$.

```
for( i=1 ; i<=20 ; i++ )
   s = s + (i *i);
```

```
i = 1;
while ( i <= 20)
{
   s = s + (i *i);
   i++;
}
```

```
i = 1;
do
{
   s = s + (i *i);
   i++;
}
while ( i <= 20);
```

Q4: Read 10 numbers, and find the sum of the positive numbers only.

```
for( i=1 ; i<=10 ; i++ )
{
   cin >> x;
   if ( x>0 )  s = s + x;
}
```

```
i = 1;
while ( i <= 10)
{
   cin >> x;
   if ( x>0 )  s = s + x;
   i++;
}
```

```
i = 1;
do
{
   cin >> x;
   if ( x>0 )  s = s + x;
   i++;
}
while ( i <= 10);
```

Q5: Represent the following series: 1, 2, 4, 8, 16, 32, 64.

```
for( i=1 ; i<65 ; i*=2 )
  cout << i;
```

```
i = 1;
while ( i<65)
{
   cout << i;
   i*=2;
}
```

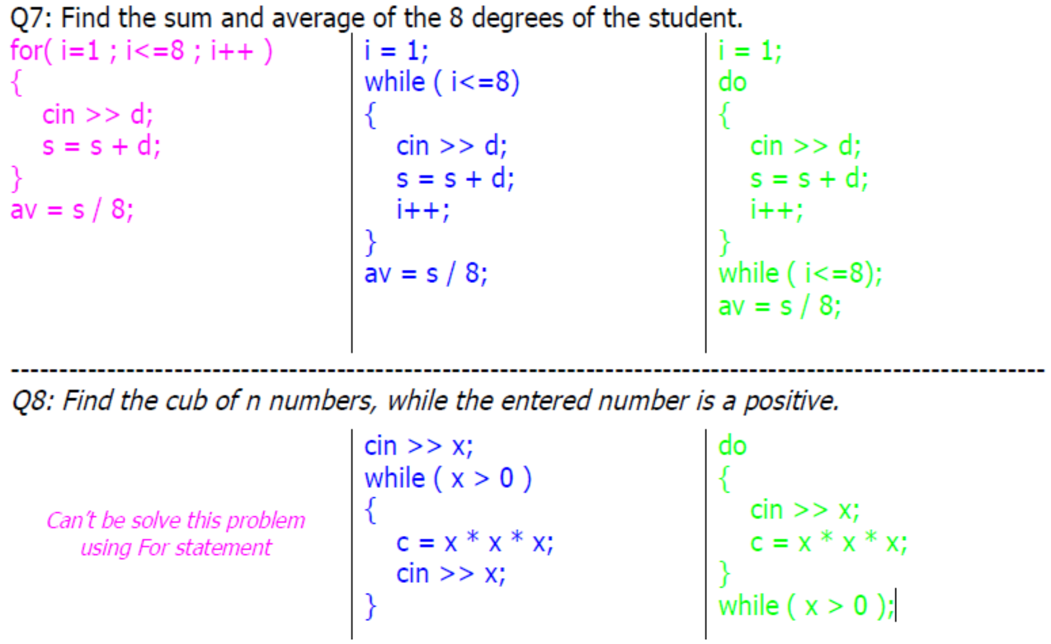
```
i = 1;
do
{
   cout << i;
   i*=2;
}
while ( i<65);
```

Q6: Find the sum of the following s = 1 + 3 + 5 + 7 + … + 99.

```
for( i=1 ; i<=99 ; i+=2 )
   s = s + i;
```

```
i = 1;
while ( i<=99)
{
   s = s + i;
   i+=2;
}
```

```
i = 1;
do
{
   s = s + i;
   i+=2;
}
while ( i<=99);
```



Q7: Find the sum and average of the 8 degrees of the student.

```
for( i=1 ; i<=8 ; i++ )
{
   cin >> d;
   s = s + d;
}
av = s / 8;
```

```
i = 1;
while ( i<=8)
{
   cin >> d;
   s = s + d;
   i++;
}
av = s / 8;
```

```
i = 1;
do
{
   cin >> d;
   s = s + d;
   i++;
}
while ( i<=8);
av = s / 8;
```

Q8: Find the cub of n numbers, while the entered number is a positive.

Can't be solve this problem using For statement

```
cin >> x;
while ( x > 0 )
{
   c = x * x * x;
   cin >> x;
}
```

```
do
{
   cin >> x;
   c = x * x * x;
}
while ( x > 0 );
```

Homework:

(1) Write ac++ program to print the following figure:

```
          5

        4 5

      3 4 5

   2  3 4  5

1  2   3  4  5
```

(2) Write ac++ program to print the following figure:

```
+

++

+++

++++

+++++

++++++

+++++++

++++++++

+++++++++

++++++++++
```

Assistant lecturer Safa Hussain

3) Write C++ program to find e from the following series:

$$e = 1 + (1/1!) + (1/2!) + (1/3!) + \dots + (1/n!)$$

4) Write C++ program to find e from the following series:

$$e = 1 + x + (x^2 / 2!) + (x^3 / 3!) + \dots (x^a / a!)$$

(5)*What is the output of the following C++ program?*

```
#include<iostream>
using namespace std;
int main( )
{int i, j, k;
for ( i = 1; i <= 2; i ++ )
{
for ( j = 1; j <= 3; j ++ )
{
for ( k = 1; k <= 4; k ++ )
cout << " & ";
cout << "\n";}
cout << "\n";
}
}
```

AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

1st STAGE STUDENTS

Assistant lecturer

Safa Hussain

Arrays

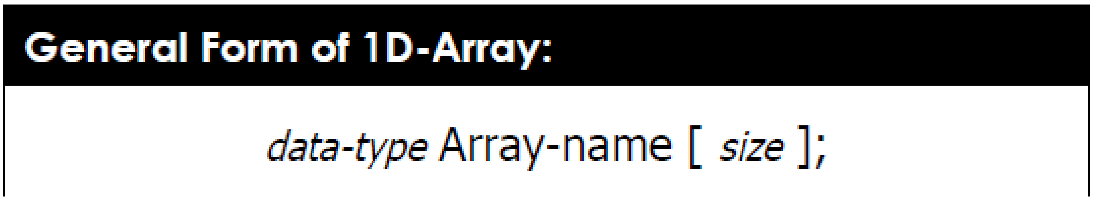
An array is a consecutive group of homogeneous memory locations.
Each element (location) can be referred to using the array name along with an integer that denotes the relative position of that element within the array. The data items grouped in an array can be simple types like int or float, or can be user-defined types like structures and objects.

Arrays are divided into:

1- Array of One Dimension
2- Array of two Dimension

Array of One Dimension:

It is a single variable specifies each array element. The declaration of one dimensional arrays is:

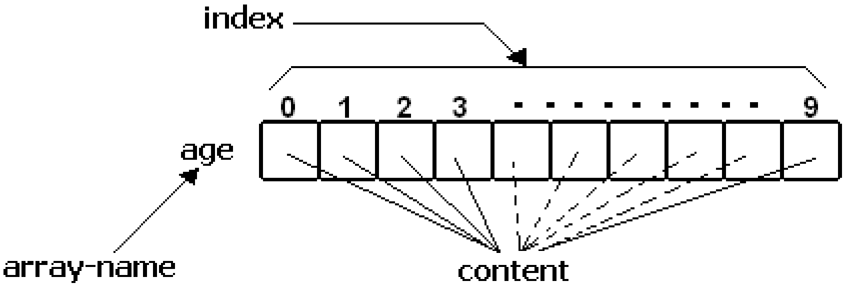


General Form of 1D-Array:

data-type Array-name [*size*];

Examples:

int age [10];
int num [30];
float degree[5];
char a [15];



The item in an array are called elements (in contrast to the items in a structure which are called members). The elements in an array are of the same type only the values vary.

To learn matrices we need four processes:

1-Array Declaration.

2-Entering Data into elements of Array.

3-Printing elements of Array.

4-Accessing to elements of Array.

Example:Write ac++ program to print (100,200)?

```
#include<iostream>
using namespace std;
int  main ( )
{
int AA[2];
 AA[0]=100;
 AA[1]=200;
cout<<AA[0]<<AA[1];
return 0;
}
```

Example:Write a c++ program to insert matrix and print matrix content?

```
#include<iostream>
using namespace std;
int  main ( )
{ int i;
int a[6];
for(i=0;i<6;i++)
    cin>>a[i];
cout<<"the content of array is \n";
for(i=0;i<6;i++)
 cout<<a[i];
return 0;
}
```

Assistant lecturer Safa Hussain

Example:Write ac++ program to insert a matrix and print reverse array order?

```
#include<iostream>
using namespace std;
int  main ( )
{
int a[6];
cout<<"enter the numbers";
for(int i=0;i<6;i++)
    cin>>a[i];
cout<<"the content of array is \n";
for(int i=5;i>=0;i--)
    cout<<a[i];
return 0;
}
```

Example:Write ac++ program to enter your name and print the characters on the reverse?

```
#include<iostream>
using namespace std;
int  main ( )
{
char a[5];
cout<<"enter the name";
for(int i=0;i<4;i++)
    cin>>a[i];
cout<<"enter the name"<<endl;
for(int i=4;i>=0;i--)
    cout<<a[i];
return 0;
}
```

Example:Write C++ program to display 2nd and 5th elements of array Distance?

```
#include<iostream>

using namespace std;

int main( )

{

double distance[ ] = { 23.14, 70.52, 104.08, 468.78, 6.28};

cout << "2nd element is: " << distance[1] << endl;

cout << "5th element is: " << distance[4];

return 0;

}
```

Example:Write C++ program to read 5 numbers and print it in reverse order?

```
#include<iostream>

using namespace std;

int main( )

{

int a [5];

cout << "Enter 5 numbers \n";

for ( int i =0; i <5; i++ )

{

cout << i << " : ";

cin >> a [i];

cout << "\n";

}

cout << "The reverse order is: \n";

for (int  i =4;i>=0;i-- )

cout << i << ": " << a [ i ] << endl;

return 0;

}
```

Example:Write C++ program, to find the summation of array elements?

```
#include<iostream>
using namespace std;
int main ( )
{
int b = 10;
int a [b];
int sum = 0;
cout << "enter 10 numbers \n";
for ( int i =0; i <b; i++ )
{
cout << "enter value " << i << ": ";
cin >> a [ i ];
sum =sum + a [ i ];
}
cout << "sum is: " << sum << endl;
return 0;
}
```

Example:Write C++ program, to find the minimum value in array of 8 Numbers?

```
#include<iostream>
using namespace std;
int  main ( )
{
   int i,a[7],min;
   cout<<"enter the seven numbers";
for ( i = 0; i < 7; i++ )
   cin>>a[i];
min=a[0];
for ( i = 0; i < 7; i++ )
if ( a [ i ] < min )
   min = a [ i ];
cout << "The minimum number in array is: " << min;
return 0;
}
```

Example:Write a c++ program To replace the third location on the first location?

```
#include<iostream>
using namespace std;
int  main ( )
{int a[10],t;
for(int i=0;i<=9;i++)
   cin>>a[i];
       t=a[0];
      a[0]=a[3];
      a[3]=t;
         for(int i=0;i<=9;i++)
         cout<<a[i];
       return 0;
          }
```

Example:Write a c++ program that input an integer array a[10] and arranges it in an ascending order?

```
#include<iostream>
using namespace std;
int  main ( )
{
int a[5];

cout<<"enter the numbers";

for(int i=0;i<5;i++)
   cin>>a[i];

for(int i=0;i<5;i++)
for(int j=i;j<5;j++)
if(a[j]<a[i])
{
   int temp=a[i];
   a[i]=a[j];
   a[j]=temp;
}
   for(int i=0;i<5;i++)
      cout<<a[i];
return 0;
}
```

Homework:

1-Write a c++ program that computes the number of even integer numbers in an array entered by the user?

2-Write a c++ program to input the elements of array a[10] and find the minimum and maximum elements?

3-Write a c++ program to input the elements of array a[10] and multiply it by 3 (a=a*3)then print the resulted array?

AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

1st STAGE STUDENTS

Assistant lecturer

Safa Hussain

Array of Two Dimension:

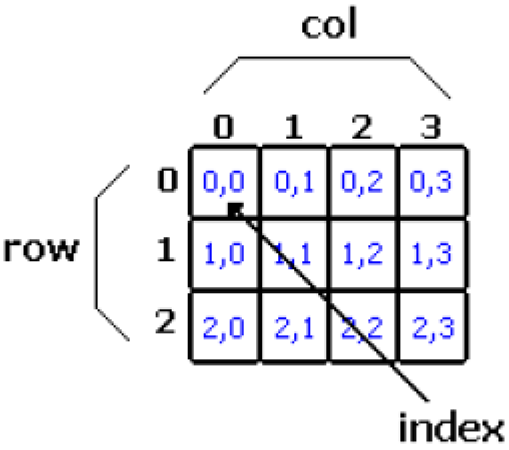
Arrays can have higher dimension. There can be arrays of two dimension which is array of arrays. It is accessed with two index. Also there can be arrays of dimension higher than two.

General Form of 2D-Array:

data-type Array-name [*Row-size*] [*Col-size*];

Examples:

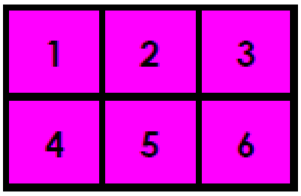
int a [10] [10];
int num [3] [4];



	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3

Initializing 2D-Array Elements:

The first element of array age:
a [2] [3] = { {1, 2, 3} , {4, 5, 6} };



1	2	3
4	5	6

Read / Write / Process Array Elements

Example1: Write C++ program, to read 15 numbers, 5 numbers per columns, and print them.

```
#include<iostream>
using namespace std;
int main ( )
{
int a [ 3 ] [ 5 ];
int i , j;
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 5; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
{
for ( j = 0 ; j < 5; j++ )
cout << a [ i ] [ j ];
cout << endl;
}
return 0;
}
```

Example2:Write C++ program, to read 4*4 2D-array, then find the summation of the array elements, finally print these elements.

```
#include<iostream>
using namespace std;
int main ( )
{
int a [ 4 ] [ 4 ];
int i , j, sum = 0;
for ( i = 0 ; i < 4; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 4; i++ )
for ( j = 0 ; j < 4; j++ )
sum += a [ i ] [ j ];
cout << "summation is: " << sum << endl;
for ( i = 0 ; i < 4; i++ )
{
for ( j = 0 ; j < 4; j++ )
cout << a [ i ] [ j ];
cout << endl;
}
return 0;
}
```

Example3: Write C++ program, to read 3*4 2D-array, then replace each value equal 5 with 0.

```
#include<iostream>

using namespace std;

int main ( )

{

int a [ 3 ] [ 4 ];

int i , j;

for ( i = 0 ; i < 3; i++ )

for ( j = 0 ; j < 4; j++ )

cin >> a [ i ] [ j ];

for ( i = 0 ; i < 3; i++ )

for ( j = 0 ; j < 4; j++ )

if ( a [ i ] [ j ] == 5 )

 a [ i ] [ j ] = 0;

for ( i = 0 ; i < 3; i++ )

{

for ( j = 0 ; j < 4; j++ )

cout << a [ i ] [ j ];

cout << endl;

}

return 0;

}
```

Example4: Write C++ program, that adds two initialized 3 * 4 matrices A and B and then stores the result in a matrix C.

```
#include<iostream>
using namespace std;
int main ( )
{
int a [ 3 ] [ 4 ], b [ 3 ] [ 4 ], c [ 3 ] [ 4 ];
int i , j;
cout << "enter element of array A: \n";
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> a [ i ] [ j ];
cout << "enter element of array B: \n";
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> b [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
c [ i ] [ j ] = a [ i ] [ j ] + b [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
{
for ( j = 0 ; j < 4; j++ )
cout << c [ i ] [ j ];
cout << endl;
}
Return 0;
}
```

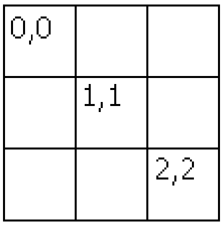
Example5: Write a c++ program that multiplies 3 * 4 matrix by 4 * 3 matrix both are entered by the user.then the program should store the result in a third matrix.

```
#include<iostream>
using namespace std;
int main()
{
   int a[3][4];
   int b[4][3];
   int c[3][3];
   cout<<"enter " <<3<<"x"<<4<<"integer matrix:"<<endl;
   for(int i=0;i<3;i++)
      for(int j=0;j<4;j++)
      cin>>a[i][j];

       cout<<"enter " <<4<<"x"<<3<<"integer matrix:"<<endl;

      for(int i=0;i<4;i++)
      for(int j=0;j<3;j++)
      cin>>b[i][j];

       for(int i=0;i<3;i++)
      for(int j=0;j<3;j++)
      {
         c[i][j]=0;
         for(int k=0;k<3;k++)
            c[i][j]+=a[i][k]*b[k][j];
      }
      cout<<"Resulted Matrix is"<<endl;
       for(int i=0;i<3;i++)
       {
```

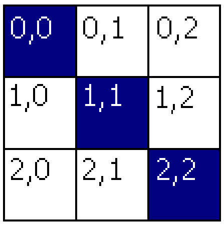
```
  for(int j=0;j<3;j++)
      cout<<c[i][j]<<" ";
          cout<<endl;

}
return 0;
}
```

Example6: Write C++ program, to replace each element in the main diameter (diagonal) with zero.

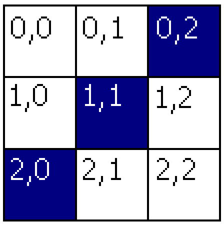
```
#include<iostream>
using namespace std;
int main ( )
{
int a [ 3 ] [ 3 ];
int i , j;
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 3; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 3; j++ )
if ( i == j )
a [ i ] [ j ] = 0;
for ( i = 0 ; i < 3; i++ )
{
for ( j = 0 ; j < 3; j++ )
cout << a [ i ] [ j ];
cout << endl;
}
return 0;}
```



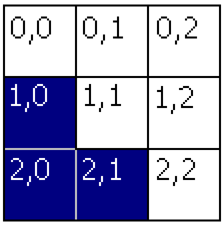
i = j



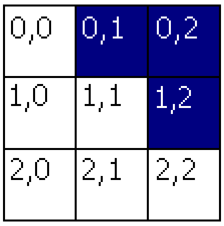
i = j



i + j = n-1



i > j



i < j

Example7:Write C++ program, to read 3*3 2D-array, then find the summation of the main diagonal and its secondary diagonal of the array elements, finally print these elements.

```
#include<iostream>
using namespace std;
int main ( )
{
int a [ 3 ] [ 3 ];
int i , j, x , y;
for ( i = 0 ; i < 3; i++ ) {
for ( j = 0 ; j < 3; j++ ) {
cin >> a [ i ] [ j ];
if ( i == j )
x=x+a[ i ][ j ];
if (i+j==4)
y=y+a[ i ][ j ];
} }
cout << "summation of diagonal is: " << x << endl;
cout << "summation of inverse diagonal is: " << y << endl;
return 0;}
```

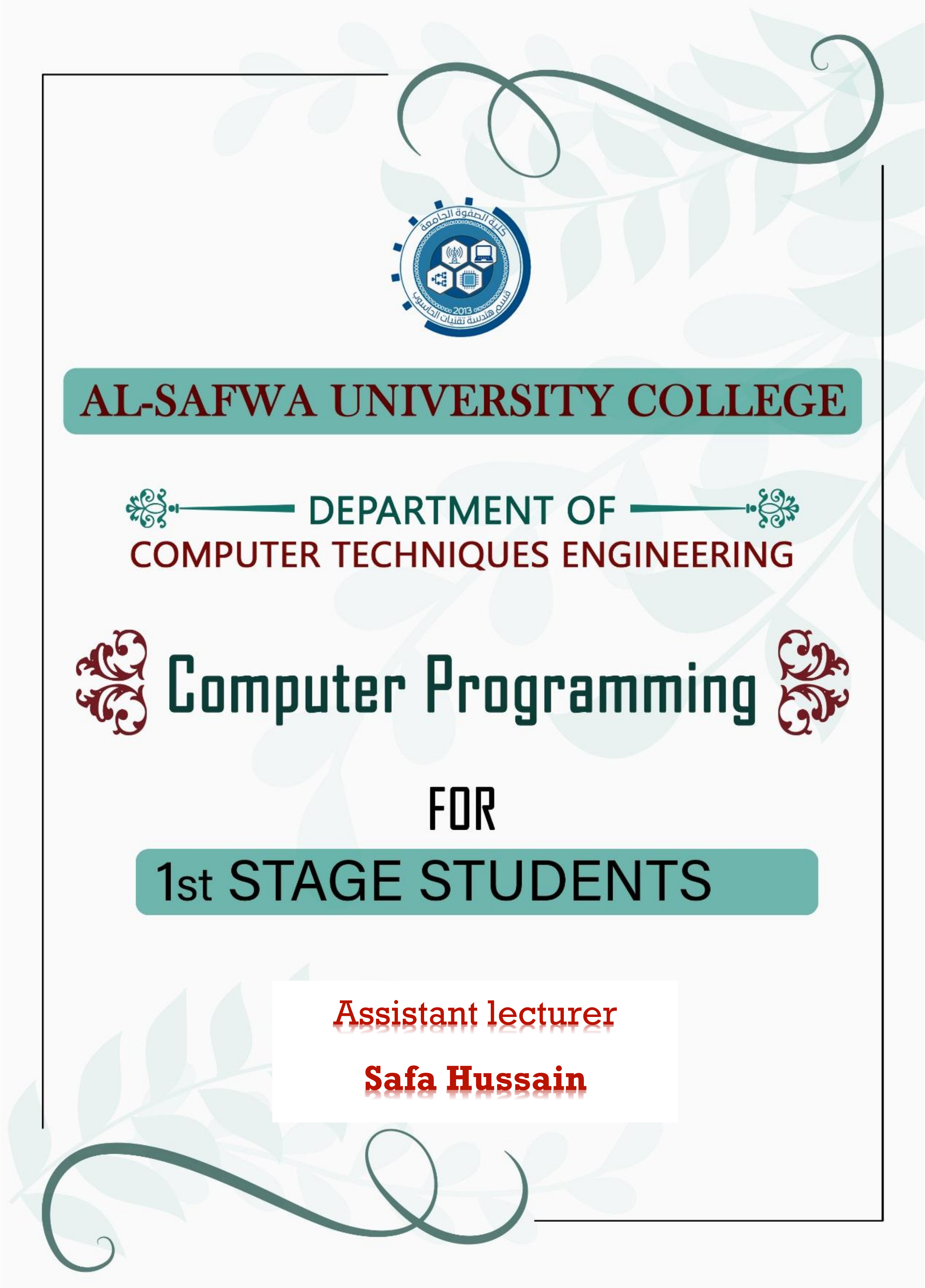
Homework:

(1) Write C++ program, to read 3*4 2D-array, then find the summation of each row.

(2) Write a c++ program that exchanges row 3 with row 1 in a 4*4 integer matrix input by the user.

(3) **Write C++ program, to read 3*3 2D-array, then find the summation of the main diagonal and its secondary diagonal of the array elements, finally print these elements.**

(4) Write C++ program, to convert 2D-array into 1D-array.



AL-SAFWA UNIVERSITY COLLEGE

DEPARTMENT OF
COMPUTER TECHNIQUES ENGINEERING

Computer Programming

FOR

1st STAGE STUDENTS

Assistant lecturer

Safa Hussain

String:

In C++ strings of characters are implemented as an array of characters. In addition a special null character, represented by \0, is appended to the end of string to indicate the end of the string.

String:-aseries of character.

General Form of String:
char String-name [*size*];

char a [11] = “Mazin Alaa”;

’M’ , ’a’ , ’z’ , ’i’ , ’n’ , ’   ’ , ’A’ , ’l’ , ’a’ , ’a’ , ’\0’

char a [ ] = “ABCD“;
’A’ , ’B’ , ’C’ , ’D’ , ’\0’

a [0] : ’A’
a [1] : ’B’
a [2] : ’C’
a [3] : ’D’
a [4] : ’\0’   null

Example:

Char x[ ]={'s','a','f','a'} it is array

```
#include<iostream>

using namespace std;

int main()

{

    char x[4]={'s','a','f','a'};

    cout<<x;

}
```

Output:

Safa ((

Example:

Char x[ ]={'s','a','f','a','\0'} it is string

```
#include<iostream>
using namespace std;
int main()
{
char x[ ]={'s','a','f','a','\0'};
    cout<<x;
}
```

Output:

safa

Note:

Function(Cin) when you found space in sentence ,it does not take the sentence after the space it is stop.

```
#include<iostream>
using namespace std;
int main()
{
   char  x[20];
    cin>>x;
    cout<<x;
}
```

Output:

```
Safa Hussain ashour
Safa
```

Note:

Function(getline) when you found space in sentence,it is take the sentence with space.

```
#include<iostream>
using namespace std;
int main()
{
   char  x[20];
    cin.getline(x,20);
    cout<<x;
}
```

Output:

```
Safa Hussain ashour
Safa Hussain ashour
```

<string.h>

It has functions ready to deal with string:

(1)Strlen

Measure the length of the string

```
Int a=strlen(s);
```

Example:

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char x[5]={'s','a','f','a','\0'};

    cout<<strlen(x);
}
```

Output:

```
4
```

Example:

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char x[50];
    cin.getline(x,50);
    cout<<strlen(x);
}
```

Output:

```
This is c++ languages
21
```

(2)strcpy

Copy content of n2 in n1

Strcpy(n1,n2);

Example:

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
   char n1[50];
   char n2[50];
   cin.getline(n1,50);
   strcpy(n2,n1);
   cout<<n2;
}
```

Output:

```
Safa Hussain
Safa Hussain
```

(3)Strcmp

Compare between 2 string

Strcmp(n1,n2) → =0 if n1=n2.
→ =1 if n1>n2.
→ =-1 if n1<n2.

Example:

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char n1[50];
    char n2[50];
    cin.getline(n1,50);
    cin.getline(n2,50);
    cout<<strcmp(n1,n2);
}
```

Output(1):

```
Computer
Computer
0
```

Output(2):

```
computer
Safa
-1
```

Output(3):

```
Am
Blue
1
```

Example:

write ac++ program that reads a string and then computes the number of capital letters in the string?

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char a[100];
    int count=0;
    cout<<"enter your string:";
    cin.getline(a,100);
    for(int i=0;a[i];i++)
        if(a[i]>='A'&&a[i]<='Z')
        count++;
    cout<<count;

}
```

Example:

use array of string to write aC++ program that prints the week days?

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
   char day[7][10]={"sunday","monday","tuesday","wednesday","thursday",
   "friday","saturday"};
   cout<<"enter your string:";

   for(int i=0;i<7;i++)


   cout<<day[i];

}
```

Homework:

(1) Write a c++ program that convert any capital letter to small in a string entered by the user?

(2) write a C++ program that computes the length of a string entered by the user?

Pointers

A pointer operator can be represented by a combination of (*) with a variable, for example int *ptr; where ptr is a pointer variable which holds the address of an integer data type.

General Form of Pointer:

Data_type *pointer_variable;

Example:

```
#include<iostream>
#include<string.h>
using namespace std;
 int main()
{
int a=10;
cout<<a;
}
```

Output:

```
10
```

Example:

```
#include<iostream>
using namespace std;
 int main()
{
int a=10;
cout<<&a;}
```

Output:

```
0x28ff0c
```

Example:

```
#include<iostream>
using namespace std;
 int main()
{
int a=10;
int *p;
b=&a;
cout<<a<<endl;
cout<<&a<<endl;
cout<<b<<endl;
cout<<*b<<endl;
}
```

Output:

```
10
0x28ff08
0x28ff08
10
```

Example:

```
#include<iostream>
using namespace std;
 int main()
{
int a[5]={1,2,3,4,5};
int*a;

cout<<a<<endl;
cout<<&a[0]<<endl;
cout<<&a[1]<<endl;
cout<<&a[2]<<endl;
cout<<&a[3]<<endl;
cout<<&a[4]<<endl;

}
```

Output:

```
0x28fefc
0x28fefc
0x28ff00
0x28ff04
0x28ff08
0x28ff0c
```