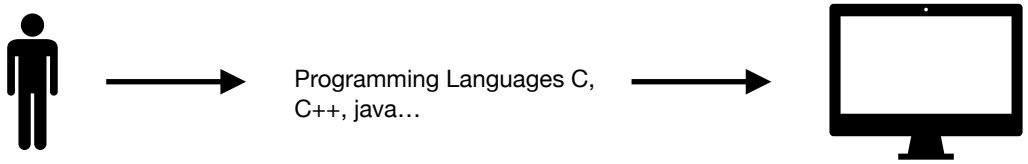
What is a Program ?

- Computer is a Programmable Computational device
- A procedure can be define as a solution steps given to computer this is called program which the CPU will execute and gives the results
- A program is a collection of **Data** and set of **Instructions**
- For learning program one should know procedures
- Data and instructions are represented as **0**'s and **1**'s in CPU as machine only understand binary data
- The intermediate language between Humana and computers is programming language which we learn in order to instruct a machine for any task



Humans

Machine

Compiler v/s Interpreter

- The main aim of theses two are

1. **To check errors**

2. **Convert into machine code**

3. **Execution / running a program** : Compiler **does not takes responsibility** of execution , Interpreter is responsible for execution of code

Compiler

- C++ is a compiler based language
- Compiler will convert program into machine code if they are no error
- Compilation from source code to machine code is done only once
- As translation is done once that why compiler doesn't run codes
- Even if there's one error in the program , it will not compile the full code must be error free to transform into machine code.

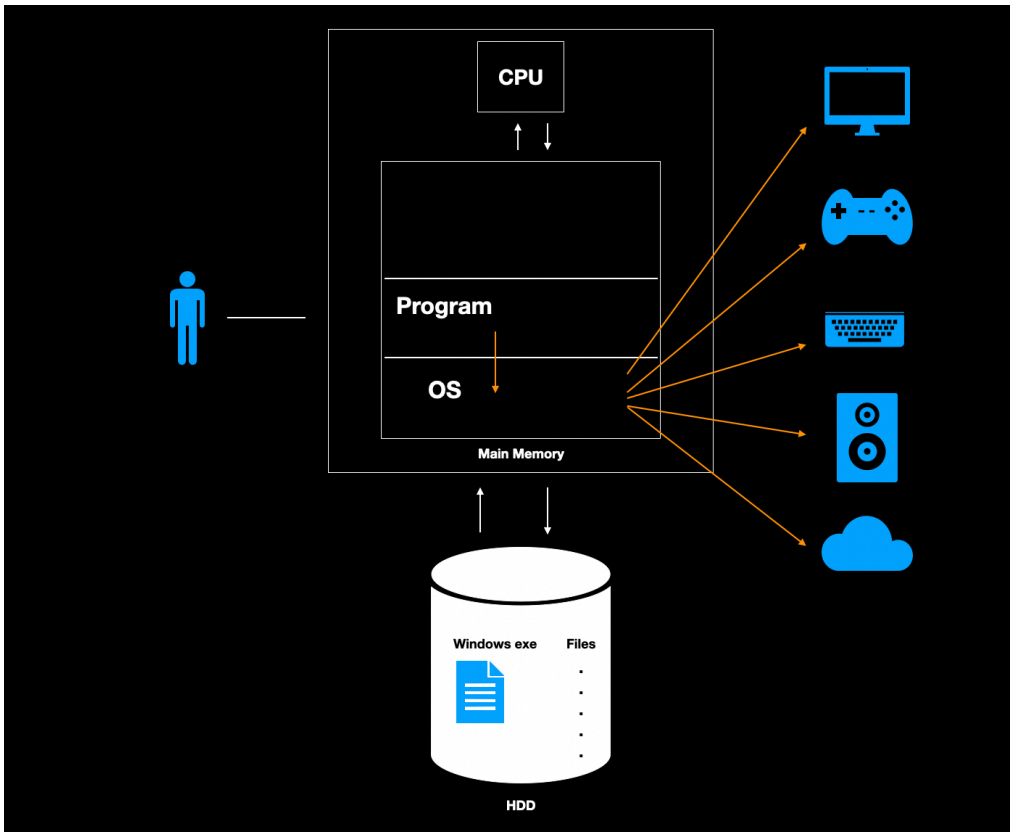
Interpreter

- JavaScript is an interpreter based language
- JS runs inside browser , it cannot run independently
- It translate and runs /executes the code line by line
- Translation is done again and again

- Compiler is faster than interpreter
- In interpreter if there is error in one line the code will still execute
- Interpreter language is easy than compiler language

Operating System

- The known operating systems are windows , linux and Mac osx , Android , iOS
- OS is a master program which provides services to the user
- It provides environment which is very user friendly
- When you first start your machine , it will get ready upon OS
- OS is also a program but it will load itself into the machine when the machine is first opened
- To use certain features OS will first load the program into the main memory and get the program ready to execute
- OS will manage all the resources , also every thing has to be done via OS i.e, for a program to avail services it should first make system call to the OS



Algorithm

- ***Algorithm*** is a step by step process for solving a computational problem
- ***programming*** is also same however , algorithms is something we write first before converting it (solution) into a program so, that our machines can understand it.
- The ***procedure*** used for solving various problems are called algorithm
- ***Pseudo code*** is something which can be written in any language (programming , syntax or pure English) this is used just for understanding algorithm

Ex of algorithm:

Algorithm Average(List , n)

{

Sun <- 0;

For each element x in list

Begin

Sum <- sum + x;

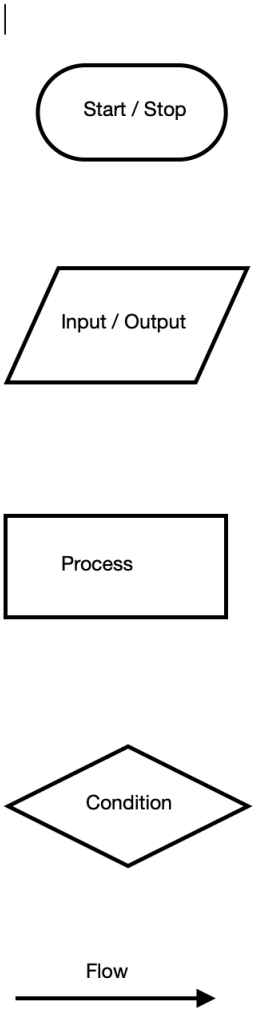
End;

Avg <- sum/n;

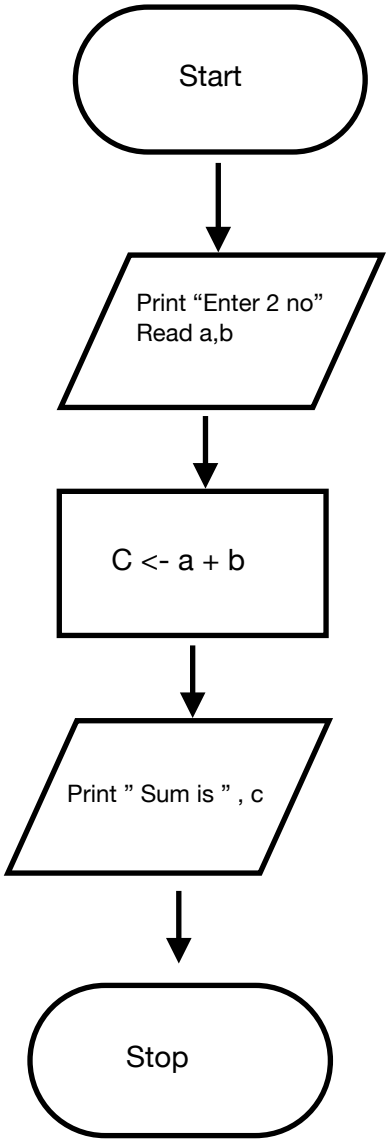
Return avg;
}

FlowChart

- Flowchart is used to show the flow of control of a program
- Flowchart is used to understand the program and its working
- It first take some ***input , process*** it and give ***output***.
- The elements of a flowchart are



- An example of Flowchart , for adding two numbers is



Steps Involve in the Development & Execution of program

- The main steps involved in the development and execution of a program are

1. Editing
2. Compiling
3. Linking Libraries
4. Loading
5. Execution

- **Editing**

- Editing can be done on any text editor / Integrated Development Environment (IDE) so, that all the above steps can be done at one place.

- **Compiling**

- Compiler will convert the source code into a machine code if there are no errors in the program

- **Linking Libraries**

- For various operations build in functions / classes are available in c++ that are present in the header files supported by libraries where machine code is readily available to use

- **Loading**

- For running the program it should be brought into the main memory from the hard disk for getting it executed this is called LOADING

- **Execution**

- Once the code is in the main memory OS will ask the CPU to execute the program thus the execution process starts.

int main() vs void main()

why we can not use void main instead of int main

some compilers supports void main and some compilers int main. Mostly int main() is used.

Why return 0;

what is the meaning or use of return 0; here and what will happen if we don't use return 0 in our code.

when a program is ending it should return 0.

It is like a standard in C++ programs, it must be writen.
return 0; means program has terminated successfully.

Why std

why we have to write std with cout. simply writing cout it will not execute?

C++ supports namespaces. All built-in functions and object are included in namespace.

There is a video available on namespaces, in ending sections.

cout is also present in namespace.
there are 2 methods of using cout.

1. std::cout
2. Using namespace std; then simply write cout.

What is #include.

is used as preprocessor directive.

#include will ask the compiler to include the header file.

there are separate videos on preprocessor directives in later sections.

Why return 0;

what is the meaning or use of return 0; here and what will happen if we don't use return 0 in our code.

when a program is ending it should return 0.

It is like a standard in C++ programs, it must be writen.
return 0; means program has terminated successfully.

In some compiler program may run without return 0; also.

How to read multiple words?

How to read multiple words in a name?

if you want to read more than one word, then use getline.

Include a header file #include<string> or #include<cstring>

getline(cin,name);

Difference between int main() and void main()

In C++ int main() is standard.

Some compiler allow void main() also.

It is mandatory to write return 0; ?

It is better to write return 0;

Some compilers may compile the program without return 0; also.

```cpp
#include<iostream>
using namespace std;

int main()
{

    string str;
    cout<<"Enetr your name";
    cin>>str;
    cout<<"Welcome "<<str;
    return 0;
}
```

What is cin.ignore()

Unable to read a string after reading a number.

I am not able to get input using getline if do sum before it. Is there any reason for that?

If your program looks like this

```
int main()
{
	int x;
	string str;
	cout<<"Enter number";
	cin>>x; // when you enter a number and hit enter
	cint>>str; // this str will take that enter key and will not read a string.
}
```

After enter a number from keyboard we hit enter key. That enter key remains in input buffer and cin>>str; will consider it as input and stops.

We should clear input buffer before reading a string .

cin.ignore() is used for clearing buffer.

Why return 0;

what is the meaning or use of return 0; here and what will happen if we don't use return 0 in our code.

when a program is ending it should return 0.

It is like a standard in C++ programs, it must be writen.
return 0; means program has terminated successfully.

In some compiler program may run without return 0; also.

How to read multiple words?

How to read multiple words in a name?

if you want to read more than one word, then use getline.

Include a header file #include<string> or #include<cstring>

getline(cin,name);

What is this endl?

endl is used for giving new line in output. Just like \n

cout<<"hi"<<"bye"; will print like this hibye

cout<<"hi"<<endl<<"bye"; will print on the screen like this

hi

bye

bye will print in next line.

Difference between variable and Object

variables are names given to values.

variable of a class is called as object.

like
int x=10;
x is of type primitive data type (int) , it is a variable.

string name="Smith";

name is a variable of type string.

string is a class in C++. So name is an object.

Why Data Types

- On data we perform required computational operations
- Data is mainly of two type

1. ***Numeric***
2. ***Character/ Alphabetic***

- Data can be either of these two or the combination go this type

- ***Numeric***

- There are represented in binary form to the machine

- They are of 2 types

I. Integer

- Integers are numbers without decimal

Ex : 2 , 1 , 56 , 45 , 23

II. Float

- The numbers with decimal are called float numbers in programming

Ex : 12.5 , 56.8 , 12.44

- ***Character/ Alphabetic***

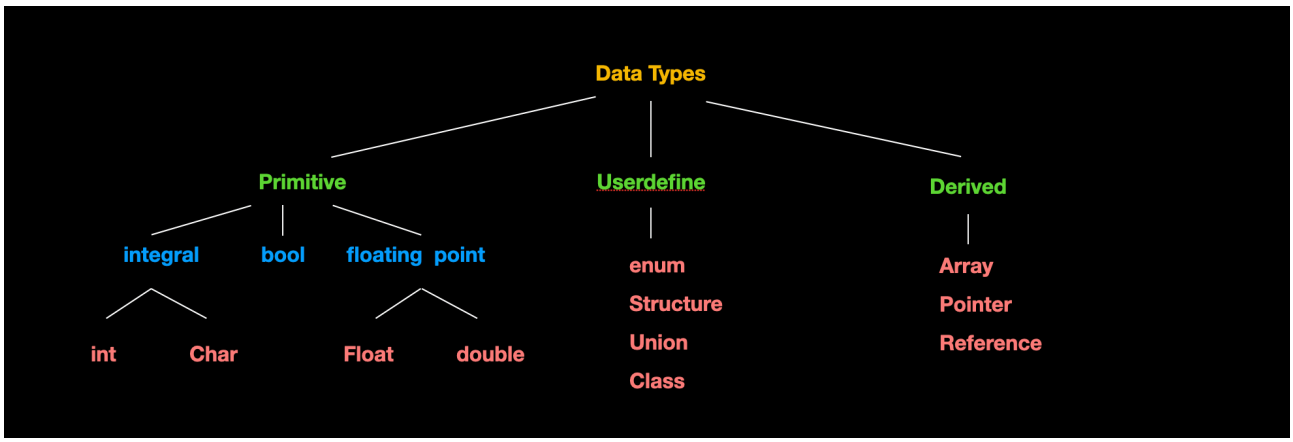
- The collection of characters or alphabets to form a single entity is called strings

- A string may be a any name , place or word from a dictionary

- c++ supports both single character or collections of characters

- There are also represented in binary form to the machine through ASCII codes

Primitive Data Types



- Data Types are called primitives data types these are directly provide by the compiler
- Primitives Data type are the basic Data Type of c++
- They are of 3 types ***integral , bool , floating point***

- **Integral**

- There is no decimal point

- **bool**

- Gives True / False as result

- **Floating Point**

- Numbers with decimal point

Sizes

Data Type	Size	Range
int	2 or 4	-32762 - 32762
float	4	-3.4×10^{-38} - 3.4×10^{38}
double	8	-1.7×10^{-308} - 1.7×10^{308}
char	1	-128 - 127
bool	undefine	True / False

Data Types and Variables

- Variables are the names given to datatypes
- Its the place where we store the data in our program
- Variables are assigned depending on the type of data
- The **integer** variables are declared as follows in c++

```
int rollno ;          // declaration variable

rollno =10 ;          // initialisation variable
```

- At runtime the variables will occupy the space in memory and the values will be stored

- Every **Character** type variable should be enclosed in ‘ ’ for instance

```
Char group = ‘A’ ;          // Declaration and initialisation of char DT
```

- The decimal numbers should be stored in **float** data type

```
float price = 12.75f ;     // declaration and initialisation of float
```

- If f is not written at the end the then it will be taken as double

How to take variable name

- Use meaningful names while declaring variable for better understanding purpose
- Some rules for declarations is as follows

int x1; 

int 1x; 

int rollno; 

int roll no; 

int roll_no; 

int rollNo; 

int RollNo; 

Operators and Expression

- Operators are predefine in c++ programming they are useful for performing some operations
- They are certain symbols used for performing operations on data in programming
- Some operators are

Arithmetic	**+ , - , * , / , %**
Relation	**< , > , <= , >= , ==**
Logical	**&& , ||, !**
Bitwise	**& , |, ~ , ^**
Increment / Decrement	**++ , - -**
Assignment	**=**

.
.
.

Arithmetic Operators :

// Example

int a , b , c;

a =10 ;

b = 5 ;

c = a + b ; **addition**
(10 + 5)

c = a - b ; **subtraction**
(10 - 5)

c = a * b ; **Multiplication**
(10 * 5)

c = a / b ; **Division (gives quotients)**
(13 / 5 = 2) // the result will be integer only
if we want result to be in decimal
then typecast it as follows

C = **(float)** a / b ;
13 / 5 = 2.6

c = a % b ; **Mod (gives remainder)**

(13 % 5 = 3)

- Mod operation cannot be performed on float numbers
- Mod is only allowed on integers and characters

sqrt() ?

For using sqrt() you have to include header file using any one method.

1. #include<cmath>

2. #include<math.h>

What is the data type of expression?

The data type of an expression will be same as the largest data type use in expression.

example:

int x=10;

float y=3;

float z=x/y; the result will be in float coz y is float.

Typecasting

If you want to change the data type of result of expression use type casting.

example:

int x=10, y=3;

float z;

z=x/y; // both x and y are int type so the result will also be int. So z=3. Even though z is float.

z=(float)x/y; //then result will be obtained in float so z=3.333.

Left and Right side of =

The data types of left and right hand side of expression are not related.

Example:

int x=10,y=3; // here both x and y are integer type

float z=x/y; // here z is float but the result of x/y will be double. Thought z is float.

float z=(float)x/y; // you have to type cast to get the result in float.

Precedence and Expression

- Expression is nothing but Formulas
- Precedence means Priority / Higher Weightage
- The priorities of the operators are as follows

Operator	Assumed Precedent
( )	3
* , / , %	2
+ , -	1

- Example :

$$X = a + b * c - d / e$$

- In this formula the first thing to be solved will be decided by the precedent according to the priority's I.e , **first * , then /**
- If you want to set the priorities then it can be done like this

$$X = (a + b) * (c - d) / e$$

- Here , the operation inside the ( ) is performed first

Precedence and Expression

- The way we can write the following Mathematic formulas in c++ is as follows with the applied condition of precedence.

Operator	Assumed Precedent
( )	3
* , / , %	2
+ , -	1

Area of Triangle - A = ½ (b × h) // Mathematics

Area of Triangle - A = 0.5 * (b * h) // C++

Perimeter of Rectangle - P = 2(l + b) // Mathematics

Perimeter of Rectangle - P = 2 * (l + b) // C++

Sum of n terms - $S = \frac{n(n+1)}{2}$ // Mathematics

Sum of n terms - S = n * (n + 1) / 2 // C++

nth term of A.P series - t = a + (n - 1) d // Mathematics

nth term of A.P series - t = a + (n - 1) * d // C++

Why return 0;

what is the meaning or use of return 0; here and what will happen if we don't use return 0 in our code.

when a program is ending it should return 0.

It is like a standard in C++ programs, it must be writen.
return 0; means program has terminated successfully.

In some compiler program may run without return 0; also.

Why return 0;

what is the meaning or use of return 0; here and what will happen if we don't use return 0 in our code.

when a program is ending it should return 0.

It is like a standard in C++ programs, it must be writen.
return 0; means program has terminated successfully.

In some compiler program may run without return 0; also.

How to read multiple words?

How to read multiple words in a name?

if you want to read more than one word, then use getline.

Include a header file #include<string> or #include<cstring>

getline(cin,name);

Difference between int main() and void main()

In C++ int main() is standard.

Some compiler allow void main() also.

It is mandatory to write return 0; ?

It is better to write return 0;

Some compilers may compile the program without return 0; also.

#include<cmath>

For using sqrt() cmath header should be included.

Why result Quadratic roots is nan ?

Following condition must be true to get real values of quadratic equation,
b^2-4*a*c>=0

Try these values of a,b, & c.

a=1 b=-4 c=4

a=1 b=-5 c=6

a=2 b=-5 c=3

a=4 b=-4 c=1

a=1 b=1 c=-6

Operators

- Various types of operators are supported by C++
- They are categorised as
- unary
- binary
- ternary
- Each operator has its precedence and associativity
- Higher precedence operators are executed first

Expression

- Expression contains operands and operator
- Expressions results in a single value
- datatype of result of expression is the largest datatype used in expression

Compound Arithmetic Operator

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

int main()
{
    int sum=10,x=5;
    sum+=x;
    cout<<sum<<endl;

    int fact=10,y=5;
    fact*=y;
    cout<<fact<<endl;
    return 0;
}
```

Program to find root of Quadratic Equation

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int a,b,c;
    float root1,root2;
    cout<<"Enter 3 values";
    cin>>a>>b>>c;
    root1=(-b+sqrt((b*b)-(4*a*c)))/(2*a);
    root2=(-b-sqrt((b*b)-(4*a*c)))/(2*a);
    cout<<root1<<" "<<root2<<endl;

    return 0;
}
```

Program to find Area of Circle

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

    float r,area;
    cout<<"Enter the Radius";
    cin>>r;
    area=3.1425f*r*r;
```

```
    cout<<"Area is "<<area<<endl;

    return 0;
}
```

Why result Quadratic roots is nan ?

Following condition must be true to get real values of quadratic equation, b^2-4*a*c>=0

Try these values of a,b, & c.

a=1 b=-4 c=4

a=1 b=-5 c=6

a=2 b=-5 c=3

a=4 b=-4 c=1

a=1 b=1 c=-6

#include<cmath>

For using sqrt() cmath header should be included.

Difference between float and double literal.

If you write a decimal value in program, like **3.57** then it is considered as double

To make it float f should be used. Like **3.57f**

Difference between int main() and void main()

In C++ int main() is standard.

Some compiler allow void main() also.

Compound Assignment Operators.

These are use for writing simple and easy to read statements.

If the statement is sum=sum+x;

same statement can be written as sum+=x;

Compound Assignment works with Arithmetic and Bitwise operators.

examples:

temp+=x;

temp-=x;

temp*=x;

temp/=x;

temp%=x;

temp<<=x;

temp>>=x;

Compound Assignment

- These are related to arithmetic and as well as other operators as well
- Usually the operators / Expression will come after the assignment but in compound assignment it comes before

Compound Assignment :

+ =

- =

*** =**

/ =

% =

& =

| =

<< =

>>=

.
.
.

Example :

int a = 10 , b = 5, c = 15

int sum = 5;

// Below , is the usual style of writing an expression

sum = sum + a

// However , the same expression can be written in compound assignment as follows

sum += a ;

The benefits of this type of programming is

- It is readable
- It is faster
- It gives more meaning

Compound Arithmetic Operator

```cpp
#include <iostream>
using namespace std;

int main()
{
    int sum=10,x=5;
    sum+=x;
    cout<<sum<<endl;

    int fact=10,y=5;
    fact*=y;
    cout<<fact<<endl;
    return 0;
}
```

Using ++ multiple times in an Expression

Multiple ++ should not be sued on a same variable inside single expression. The result will be unpredictable. The result is dependent on compiler.

Example:

```
int i=5,j;

j=++i + i++ + i++; //here we cant say what the result unless we run the program


cout<<i++<<++i<<i++; //here we cant say what will be output unless we ruin the program.
```

Increment and Decrement Operator

- Increment and Decrement are of two types I.e pre increment, post increment and pre decrement ,post decrement

Example :

Pre inc : ++x ;

Post inc : x++;

Pre dec : ++x ;

Post dec : x++;

- These variables are useful for counting as they are very common so c++ has introduced this concept

Example :

i = i + 1;

- This statement can be written as

i +=1 ;

i++;

- This is used mostly in loops

- The main difference between pre increment and post increment is, although they both give the same result

- The meaning of pre increment is, first increment the value then assign the result to the variable

Ex:

Int x = 5, y ;

y = ++ x;

- The meaning of post increment is first assign then increment it the value.

Ex:

Int x = 5, y ;

y = x ++ ;

Increment ans Decrement Operators

```cpp
#include <iostream>
using namespace std;

int main()
{
    int i=5,j;
    j=i++;
    cout<<j<<" "<<i<<endl;

    int k=5,l;
    l=++k;
    cout<<l<<" "<<k<<endl;

    int a=5,b;
    b=2*++a + 2*a++;
    cout<<b<<" "<<a<<endl;;

    int c=5,d;
    d=2*c++ + 2*c++;
    cout<<d<<" "<<c<<endl;

    return 0;
}
```

How Negative Numbers are stored.

Negative numbers are in 2's Compliment form. Let lost bit will be always 1

example:

POSITIVE

char x=5;

Binarny form of x will be 0000 0101

NEGATIVE

char x=-5

Binary form of 5 is 0000 0101

1's Compliment is 1111 1010

2's Compliment is 1111 1011 (2's comp = 1's comp + 1)

~ Operator

Operator will invert the bits of a variable

char x=5

Binary form is 0000 0101

x=~x will be 1111 1010

Left most bit is 1 so it is a negative number

Finding the value of 1111 1010

1's compliment will be 0000 0101

2's compliment will be 0000 0110 = 6

Therefor 1111 1010 is a -6

Bitwise operators don't work on float

Bitwise operators works only on int and char type

Overflow

- When the value is more than the capacity it will take the values again from the beginning this concept is called overflow

- Suppose there is a byte and we are already having values in it , if 1 is added to this max byte value it becomes a signed bit therefor it become -128 magically this happens when we try going to the next value

- The number we got is in twos compliment as -ve numbers are stored in twos compliments, if we want to find the original number we should again find its two compliment

Program to Demonstrate Overflow

```
#include <iostream>
using namespace std;
int main()
{
    char a=128;
    cout<<(int)a<<endl;

    char b=127;
    b++;
    cout<<(int)b<<endl;

    char c=-129;
    cout<<(int)c<<endl;

    char d=-128;
    d--;
    cout<<(int)d<<endl;

    int e=INT_MAX;
    e++;
    cout<<(int)e<<endl;

    return 0;
}
```

How Negative Numbers are stored.

Negative numbers are in 2's Compliment form. Let lost bit will be always 1

example:

POSITIVE

char x=5;

Binarny form of x will be 0000 0101

NEGATIVE

char x=-5

Binary form of 5 is 0000 0101

1's Compliment is 1111 1010

2's Compliment is 1111 1011 (2's comp = 1's comp + 1)

~ Operator

Operator will invert the bits of a variable

char x=5

Binary form is 0000 0101

x=~x will be 1111 1010

Left most bit is 1 so it is a negative number

Finding the value of 1111 1010

1's compliment will be 0000 0101

2's compliment will be 0000 0110 = 6

Therefor 1111 1010 is a -6

Bitwise operators don't work on float

Bitwise operators works only on int and char type

Bitwise Operator

- These operators are performed on the bits of the data and not as the single whole data unit
- The operations available here are

& - and

| - or

^ - xor

~ - not

<< - left shift

>> - Right Shift

- Its working

bit 1	bit 2	bit 1 & bit 2
0	0	0
1	0	0
0	1	0
1	1	1

bit 1	bit 2	bit 1 | bit 2
0	0	0
1	0	1
0	1	1
1	1	1

bit 1	bit 2	bit 1 ^ bit 2
0	0	0
1	0	1
0	1	1
1	1	0

- If you are making device drivers , systems program, System application , core system program and in general if a person is working more close towards electronic there this operator is used

- Left and Right shift

int x = 5 , y;

X = 0 0 0 0 0 1 0 1

0 0 0 0 0 1 0 1

- When you shift all the bits on left hand side by one space then 5 will get multiplied by 2 , if you move them by 2 space it will be multiplied by 4

- Here signed bit is not included , if the number is positive then it will be positive and vice versa

Bitwise Operators

Program to Demonstrate bitwise operators

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

int main()
{
    int a=11,b=7,c;
    c=a&b;
    cout<<c<<endl;

    int d=11,e=7,f;
    f=d|e;
    cout<<f<<endl;

    int g=11,h=7,i;
    i=g^h;
    cout<<i<<endl;

    char j=5,k;
    k=j<<1;
    cout<<(int)k<<endl;

    char l=20,m;
    m=l>>1;
    cout<<(int)m<<endl;

    char x=5,y;
    y= ~ x;
    cout<<(int)y<<endl;

    return 0;
}
```

Enumerated and Typedef

- Enum or enumerated data type is used to define user-define datatype
- With the help of existing data type we can define our own data type this is called Enum
- To make the work faster codes are given to the words it is a common practise among us as well
- That code can be used in our programming ti represent that particular word
- If they are a lot of numbers then they can be grouped together under one name that is enum
- They method of defining enum is

Ex :

enum days { mon , tue , wed , thurs , fri , sat , sun };

0 1 2 3 4 5 6

- Here you can access the values that are assigned to it already you cannot define your own values here

Ex:

- By default the index value assigned is 0 but you can change it if you want

enum dept { CS, IT , EEE , C IVIL };

1 2 3 4

- This feature and its usage is completely upto the programmer

Typedef :

- To give meaning full names to the variables typedef is used
- It is also used to define user define data type
- It is used for better readability of a program

Ex :

```
typedef int marks ;
typedef int rollno ;

Int main( )
{
marks m1, m2, m3;

rollno r1 ,r2, r3;
}
```

Program to Calculate Area of Circle

```
#include<iostream>
using namespace std;
int main()
{
    float radius;
    float area;

    cout<<"Enrer Radius of a Circle";
    cin>>radius;

    area=3.1425*radius*radius;

    cout<<"Area of a Circle is "<<area<<endl;

}
```

Program to Calculate Net Salary

```
#include<iostream>
using namespace std;
int main()
{
    float basic;
    float percentAllow;
    float percentDeduct;
    float netSalary;

    cout<<"Enter Basic Salary:";
    cin>>basic;
    cout<<"Enter percent of Allowences:";
    cin>>percentAllow;
    cout<<"Enter percent of Deductions:";
    cin>>percentDeduct;

    netSalary=basic+basic*percentAllow/100-
basic*percentDeduct/100;

    cout<<"Net Salary is:"<<netSalary<<endl;

}
```

Conditional Statements

- If and else is used for writing conditional statement
- If condition is true then if block is executed
- If condition is false then else block is executed
- 0 - means false
- 1- means true or non0zero value is also true

If can be nested inside if as well as else statement
Nesting of is is also written as else-if ladder

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Conditional Statements

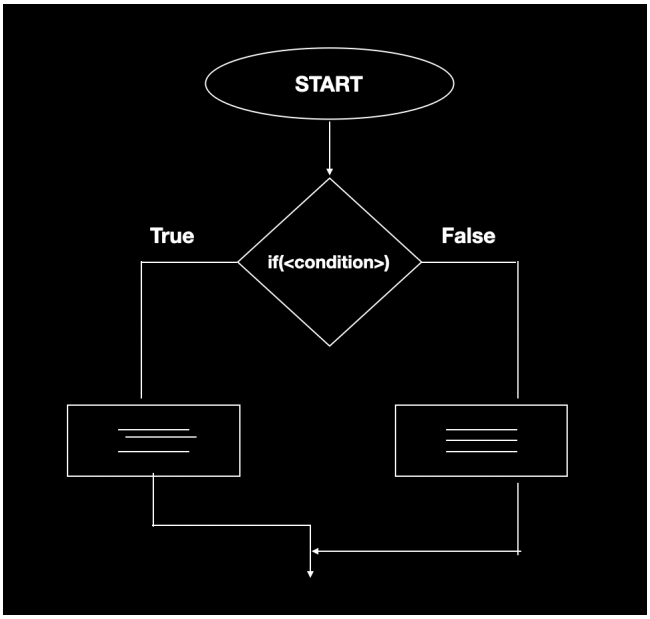
- In daily life we come across conditional problems for instance if you are Loggin in somewhere the you need to enter correct password in order to login otherwise no login

- The skeleton for conditional statement is

```
If (<condition>)

{
        ……..
        ……..
        ……..

}

else
{
        ……..
        ……..
        …….

}
```

- Depending on the result of the condition if block is executed otherwise else will be executed

The flow of conditional statement is given below



- The statement is considered False if the value is 0 (zero) , any other value other than 0 is True
- The condition is written using relational operator (< , > <=, >= , == ,!=) as they are used to compare the two values we use relational operator

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

/*program for finding maximum of 2 nos using conditional statements


*/
int main()
 {

      int x,y;

      cout<<"enter 2 nos";
      cin>>x>>y;
      if(x>y)
      {
            cout<<"maximum is"<<x;
      }
      else
      {
            cout<<"maximum is"<<y;
      }
      return 0;
 }
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Program to Demonstrate Simple Conditional Statement

```cpp
#include <iostream>
using namespace std;

int main()
{
    int roll;
    cout<<"Enter your Roll No."<<endl;
    cin>>roll;
    if(roll>0)
    {
        cout<<"Valid Roll No."<<endl;
    }
    else
    {
        cout<<"Invalid Roll No."<<endl;
    }
    return 0;
}
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Program to perform Validation

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

int main()
{
    int a,b,c;
    cout<<"Enter two numbers: "<<endl;
    cin>>a>>b;
    if(b==0)
    {
        cout<<"Invalid denominator"<<endl;
    }
    else
    {
        c=a/b;
        cout<<c<<endl;
    }

    return 0;
}
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Logical Operators

- If you are having more than one conditional statements and wants to join them, then this concept is called compound conditional statement and the operations using which this can be done is called Logical Operator

Logical Operators are :

&&	AND
||	OR
!	NOT

- && , || is used to make compound conditional statements
- while ! Is used for negations that is make a true statement false and vice versa
- The values of AND , OR depends on their truth values because they are logical

&& - AND

T-shirt	Cap	T-shirts AND Cap
T	T	T
T	F	F
F	T	F
F	F	F

- In AND if one is FALSE everything is FALSE

|| - OR

- In OR if one is TRUE everything is TRUE

T-shirt	Cap	T-shirts OR Cap
T	T	T
T	F	T
F	T	T
F	F	F

! - NOT

T-shirt	! T-shirt
T	F
F	T

Program to Demonstrate Compound Conditional Statement

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

int main()
{
    int age;
    cout<<"Enter your age: "<<endl;
    cin>>age;

    if(age>=12 && age<=50)
    {
        cout<<"Young"<<endl;
    }
    else
    {
        cout<<"Not Young"<<endl;
    }

    if(age<12 || age>50)
    {
        cout<<"Eligible for the offer"<<endl;
    }
    else
    {
        cout<<"Not eligible for the offer"<<endl;
    }

    return 0;
}
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

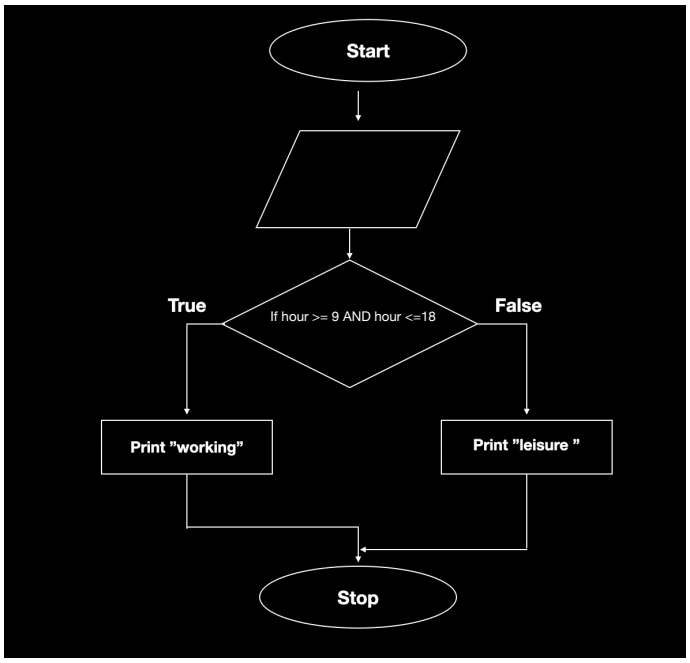
in switch case. for the last case or default, break is not necessary. It can be skipped

Compound Conditional Statement

- We can combine more than one conditional statement using AND OR
- We use compound conditional statement when we want to check something within range and since range consists 2 values we need something to join ignorer to check it hence we use AND , OR statements
- **Example :**

```
int main( )
{

        int hour;
        cout<< “Enter Hours ”;
        cin>> hour;
        if(hour >= 9  && hour <= 18 )
{
        count<< “Working ” ;
}

else
{
        count<< “ Leisure ” ;
}
Return 0;
}
```

- The flow of this compound conditional statements is as follows



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

/*program for finding maximum of 2 nos using and compound conditional
statements


*/
int main()
 {

      int age;
      cout<<"enter age";
      cin>>age;
      if(age>=12&&age<=50)
      {
            cout<<"young";
       }
            else
      {
            cout<<"not young";
       }
      return 0;
 }
```

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

/*program for finding a person eligible or not using or compound
conditional statements


*/
int main()
 {

      int age;
      cout<<"enter age";
      cin>>age;
      if(age<=12||age>=50)
      {
            cout<<"eligible";
       }
            else
      {
            cout<<"not eligible";
       }
      return 0;
 }
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Program to Find Greatest of 3 Numbers

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

int main()
{
    int a,b,c;
    cout<<"Enter 3 no.s"<<endl;
    cin>>a>>b>>c;
    if(a>b && a>c)
    {
        cout<<a<<endl;
    }
    else if(b>c)
    {
        cout<<b<<endl;
    }
    else
    {
        cout<<c<<endl;
    }
    return 0;
}
```

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

/*program for find nature of quadratic roots


*/
int main()
 {

      float a,b,c,d,r1,r2;
      cout<<"enter a,b,c";
      cin>>a>>b>>c;
      d=b*b_4*a*c;

      if(d==0)
      {
            cout<<"roots are real and equal"<<endl;
            cout<<(_b/(2*a));
       }
            else if(d>0)
            {

                  cout<<"roots are real and unequal"<<endl;
                  cout<<(_b+sqrtd/(2*a));
                  cout<<(_b_sqrtd/(2*a));
            }
                  else
                  cout<<"imaginery";
                  return 0;

}
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

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

/*program for displaying day name using else if ladder


*/
int main()
 {
      int day;
        cout<<"enter the day number";
        cin>>day;
        if(day==1)
           cout<<"monday"<<endl;
        else if(day==2)
             cout<<"tuesday"<<endl;
        else if(day==3)
             cout<<"wednesday"<<endl;
        else if(day==4)
             cout<<"thursday"<<endl;
        else if(day==5)
           cout<<"friday"<<endl;
        else if(day==6)
             cout<<"saturday"<<endl;
        else if(day==7)
             cout<<"sunday"<<endl;
        else
             cout<<"invalid day no"<<endl;
        return 0;
}
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Program to Demonstrate Short Circuit

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

int main()
{
    int a=10,b=5,i=5;

    if(a>b && ++i<=b)
    {
    }
    cout<<i<<endl;

    if(a<b || ++i<=b)
    {
    }
    cout<<i<<endl;
}
```

Dynamic Declaration

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

int main()
{
    int a=10,b=5;

    if(true)
    {
        int c=a+b;
        cout<<c<<endl;
    }

    {
    int d=a-b;
    if(true)
    {
        cout<<d<<endl;
        }
    }

    if(int e=a*b)
    {
        cout<<e<<endl;
    }
}
```

Switch

- Switch is a branch and control statement
- Switch can have 0 or more cases
- Each case is defined with a label
- Depending on the value of expression in switch corresponding case black is executed
- Is a case block is not available then default block is executed
- Default block is optional
- Every case block must terminate with break
- If breaks are not mentioned then cases will fall thru
- Switch is used for menu-driven programs

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

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

/*program for displaying day name using switch case


*/
int main()
{
      int day;
      cout<<"enter the day no";
      cin>>day;
        switch(day)
        {
             case 1: cout<<"monday"<<endl;
                  break;
           case 2: cout<<"tuesday"<<endl;
                break;
             case 3: cout<<"wednesday"<<endl;
              break;
           case 4: cout<<"thursday"endl;
              break;
           case 5: cout<<"friday"<<endl;
              break;
           case 6: cout<<"saturday"<<endl;
              break;
           case 7: cout<<"sunday"<<endl;
              break;
           default: cout<<"invalid day no"<<endl;
      }
      return 0;
}

------------------------------------------------------------

# include<iostream>
using namespace std;

/*switch case program using menu driven options


*/
int main()
{
      cout<<"menu\n";
      cout<<"1. add\n"<<"2. sub\n"<<"3. multi\n"<<"4. div\n";
         int option;
            cout<<"enter your choice";
            cin>>option;
         int a,b,c;
            cout<<"enter 2 numbers";
            cin>>a>>b;
      switch(option)
          {
            case 1:c=a+b;
                     break;
            case 2:c=a_b;
               break;
```

```
            case 3:c=a*b;
               break;
            case 4:c=a/b;
               break;
        }
            cout<<"result is"<<c<<endl;
            return 0;
}
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Program to Calculate Discounted Bill Amount

```
#include<iostream>
using namespace std;
int main()
{
    float billAmount;
    float discount=0.0;

    cout<<"Enter Bill Amount:";
    cin>>billAmount;
    if(billAmount>=500)
        discount=billAmount*20/100;
    else if(billAmount>=100 && billAmount<500)
        discount=billAmount*10/100;

    cout<<"Bill Amount is:"<<billAmount<<endl;
    cout<<"Discount is :"<<discount<<endl;
    cout<<"Discounted Amount is:"<<billAmount-
discount<<endl;


}
```

Program to to check if its a Leap Year

```cpp
#include <iostream>

using namespace std;


int main()
{
    int year;

    cout << "Enter a year: ";
    cin >> year;

    if (year % 4 == 0)
    {
        if (year % 100 == 0)
        {
            if (year % 400 == 0)
                cout << year << " is a leap year.";
            else
                cout << year << " is not a leap year.";
        }
        else
            cout << year << " is a leap year.";
    }
    else
        cout << year << " is not a leap year.";

    return 0;
}
```

True vs False

In C++ true and false are defined as

0 means False

Another value is True. It may be 1 ,2 ,10, 1.5, 100 etc.

ASCII codes

Characters in C++ are represented using inter codes called as ASCII codes.

Character will actually store number. As it is declared as char, it represents characters.

Every character you find on KeyBoard will have its ASCII codes

ASCII codes for Capital letters range from A=65 to Z=90

ASCII codes for Lower case letter range from a=97 to z=122

Difference between Upper case and Lower case is 32

Arithmetic operations can be performed on char. It will change ASCII code

break vs return vs exit(0);

break statement will stop the loop or switch case

return statement will stop the function

exit(0) will stop the program

Dynamic Declaration of variables in C

Yes in latest compilers of C dynamic declaration is allowed.

Block level Scope of Variable

if a variable is declared inside a block, then it will be deleted from the memory at the end of block

example:

if() { int x; }

while(){ int x; }

int fun(){ int x; }

switch case vs if-else

If-else ladder will check multiple conditions and then execute the block if condition is true

Switch statement will directly jump to the case block.

If-else ladder is slow and switch is fast.

Switch can be used only for checking single value. If-else can be used for checking range of values.

break after default

in switch case. for the last case or default, break is not necessary. It can be skipped

Loops

- Loops are iterative statements
- Block of statements are repeatedly executed as long as condition is true
- Condition given in loop must become false after some finite iterations otherwise its a infinite loop
- Values used in condition must update inside the body of finite loop
- Four types of loops
-
- pre-tested loop while()
- post-tested loop do..while()
- counter controlled loop for()
- for each loop for Collections for()

while and for difference

while is used when you don't know how many time you have to repeat, so repeat WHILE condition is true.

for is used when you know FOR how many time you have to repeat.

When to use which loop ?

1. While and doWhile are similar, they are used when we don’t know number of times of repetitions

2. For is used when we know the number of iterations.

3. For each is used with array or STL.

What should be used in for loop i++ or ++i ?

Anyone can be used.

++i; is known to be faster than i++;.

Reason:

i++; is same as i=i+1; here i+1 is evaluated and the result is stored in a temporary variable and then assigned to i.

++i; here i+1 is not stored in a temporary variable, directly i is increased by 1

Factorial for large numbers

if the data type is taken long long then factorial will work upto 20!

if you want beyond 20! then take a string and write own procedure to calculate factorial.

Armstrong Number, why number is stored in m ?

actual number is taken in n.

While loop will stop when n=0

We have to compare sum with the number, so it is already stored in m.

Reverse a number ending with zero.

if a number is ending with 0 and you want 0 also then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';
cout<<reverse;
```

Display a number in words.

if a number is ending with 0 and you want to display in words then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';

for(int i=0;reverse[i]!='\0';i++)
{
	switch(reverse[i]-'0')
	{
		case 0: cout<<"Zero ";
			break;
		case 1: cout<<"One ";
			break;
		case 2: cout<<"Two ";
			break;
		.
		.
		.
	}
}
```

while and for difference

while is used when you don't know how many time you have to repeat, so repeat WHILE condition is true.

for is used when you know FOR how many time you have to repeat.

When to use which loop ?

1. While and doWhile are similar, they are used when we don’t know number of times of repetitions

2. For is used when we know the number of iterations.

3. For each is used with array or STL.

What should be used in for loop i++ or ++i ?

Anyone can be used.

++i; is known to be faster than i++;.

Reason:

i++; is same as i=i+1; here i+1 is evaluated and the result is stored in a temporary variable and then assigned to i.

++i; here i+1 is not stored in a temporary variable, directly i is increased by 1

Factorial for large numbers

if the data type is taken long long then factorial will work upto 20!

if you want beyond 20! then take a string and write own procedure to calculate factorial.

Armstrong Number, why number is stored in m ?

actual number is taken in n.

While loop will stop when n=0

We have to compare sum with the number, so it is already stored in m.

Reverse a number ending with zero.

if a number is ending with 0 and you want 0 also then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';
cout<<reverse;
```

Display a number in words.

if a number is ending with 0 and you want to display in words then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';

for(int i=0;reverse[i]!='\0';i++)
{
	switch(reverse[i]-'0')
	{
		case 0: cout<<"Zero ";
			break;
		case 1: cout<<"One ";
			break;
		case 2: cout<<"Two ";
			break;
		.
		.
		.
	}
}
```

Program to Test all Loops

```cpp
#include <iostream>
using namespace std;

int main()
{
    int a=0;
    while(a<10)
    {
        cout<<"a "<<a;
        a++;
    }

    int b=0;
    do
    {
        cout<<"b "<<b;
        b++;
    }while(b<10)

    ;for(int i=0;i<10;i++)
    {
        cout<<"i "<<i;
    }

    return 0;
}
```

Program for Infinite Loop

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

int main()
{
    int i=0;
    ;for(;;)
    {
        cout<<" Hello";
        i++;
    }
}
```

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

/*program for multiplication table using for loop


*/
int main()
{
      int n,i;
      cout<<"enter a number";
      cin>>n;
      for(i=1;i<=10;i++)
      {
          cout<<n<<" x "<<i<<" = "<<i*n<<endl;
      }
      return 0;
}
```

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

/*program for sum of natural numbers using for loops

*/
int main()
{
      int n,i,sum=0;
      cout<<"enter the number";
      cin>>n;
      for(i=1;i<=n;i++)
      {
          sum=sum+i;
      }
      cout<<"sum of natural no is"<<sum<<endl;
      return 0;
}
```

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

/*program for sum of natural numbers using while loops


*/
int main()
{
      int n,i=1,sum=0;
      cout<<"enter the number";
      cin>>n;
      while(i<n)
      {
           sum=sum+i;
           i++;
```

```
        }
        cout<<"sum of natural no is"<<sum<<endl;
        return 0;
}
```

while and for difference

while is used when you don't know how many time you have to repeat, so repeat WHILE condition is true.

for is used when you know FOR how many time you have to repeat.

When to use which loop ?

1. While and doWhile are similar, they are used when we don’t know number of times of repetitions

2. For is used when we know the number of iterations.

3. For each is used with array or STL.

What should be used in for loop i++ or ++i ?

Anyone can be used.

++i; is known to be faster than i++;.

Reason:

i++; is same as i=i+1; here i+1 is evaluated and the result is stored in a temporary variable and then assigned to i.

++i; here i+1 is not stored in a temporary variable, directly i is increased by 1

Factorial for large numbers

if the data type is taken long long then factorial will work upto 20!

if you want beyond 20! then take a string and write own procedure to calculate factorial.

Armstrong Number, why number is stored in m ?

actual number is taken in n.

While loop will stop when n=0

We have to compare sum with the number, so it is already stored in m.

Reverse a number ending with zero.

if a number is ending with 0 and you want 0 also then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';
cout<<reverse;
```

Display a number in words.

if a number is ending with 0 and you want to display in words then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';

for(int i=0;reverse[i]!='\0';i++)
{
	switch(reverse[i]-'0')
	{
		case 0: cout<<"Zero ";
			break;
		case 1: cout<<"One ";
			break;
		case 2: cout<<"Two ";
			break;
		.
		.
		.
	}
}
```

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

/*program for finding factorial of a number


*/
int main()
{
      int n,i=1,fact=1;
      cout<<"enter the number";
      cin>>n;
      for(i=1;i<=n;i++)
      {
            fact=fact*i;
      }
      cout<<"factorial of"<<n<<"is"<<fact<<endl;
      return 0;
}
```

while and for difference

while is used when you don't know how many time you have to repeat, so repeat WHILE condition is true.

for is used when you know FOR how many time you have to repeat.

When to use which loop ?

1. While and doWhile are similar, they are used when we don’t know number of times of repetitions

2. For is used when we know the number of iterations.

3. For each is used with array or STL.

What should be used in for loop i++ or ++i ?

Anyone can be used.

++i; is known to be faster than i++;.

Reason:

i++; is same as i=i+1; here i+1 is evaluated and the result is stored in a temporary variable and then assigned to i.

++i; here i+1 is not stored in a temporary variable, directly i is increased by 1

Factorial for large numbers

if the data type is taken long long then factorial will work upto 20!

if you want beyond 20! then take a string and write own procedure to calculate factorial.

Armstrong Number, why number is stored in m ?

actual number is taken in n.

While loop will stop when n=0

We have to compare sum with the number, so it is already stored in m.

Reverse a number ending with zero.

if a number is ending with 0 and you want 0 also then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';
cout<<reverse;
```

Display a number in words.

if a number is ending with 0 and you want to display in words then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';

for(int i=0;reverse[i]!='\0';i++)
{
	switch(reverse[i]-'0')
	{
		case 0: cout<<"Zero ";
			break;
		case 1: cout<<"One ";
			break;
		case 2: cout<<"Two ";
			break;
		.
		.
		.
	}
}
```

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

/*program for finding factors of a number


*/
int main()
      int i,n;
      cout<<"enter the number";
      for(i=1;i<=n;i++)
      {
           if(n%i==0)
           {
            cout<<i<<endl;
           }
      }
      return 0;
}
```

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

/*program for finding perfect number


*/
int main()
{
      int i,n;
      cout<<"enter the number";
      for(i=1;i<=n;i++)
      {
           if(n%i==0)
           {
            sum=sum+i;
           }
      }
        if(2*n==sum)
            cout<<"perfect number";
        else
            cout<<"not a perfect number";
      return 0;
}
```

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

/*program for finding sum of the factors of a number


*/
int main()
{
```

```
      int i,n;
      cout<<"enter the number";
      for(i=1;i<=n;i++)
      {
           if(n%i==0)
           {
            sum=sum+i;
           }
      }
      cout<<"sum of factors"<<sum<<endl;
      return 0;
}
```

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

/*program for finding prime number


*/
int main()
{
      int i,n;count=0;
      cout<<"enter the number";
      for(i=1;i<=n;i++)
      {
           if(n%i==0)
           {
            count++;
           }
      }
      if(count==2)
            cout<<"prime number";
      else
            cout<<"not a prime number";
      return 0;
}
```

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

/*program for display digits of a number reverse


*/
int main()
{
      int n,r;
      cout<<"enter the number";
      cin>>n;
      while(n>0)
      {
            r=n%10;
              r=n/10;
```

```
            cout<<r<<endl;
      }
      return 0;
}
```

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

/*program for finding armstrong number


*/
int main()
{
      int n,r,sum=0,m;
      cout<<"enter the number";
      cin>>n;
      m=n;
      while(n>0)
      {
            r=n%10;
               r=n/10;
            sum=sum+r*r*r;
      }
      if(sum==m)
            cout<<"armstrong number";
      else
            cout<<"not a armstrong number";
      return 0;
}
```

while and for difference

while is used when you don't know how many time you have to repeat, so repeat WHILE condition is true.

for is used when you know FOR how many time you have to repeat.

When to use which loop ?

1. While and doWhile are similar, they are used when we don’t know number of times of repetitions

2. For is used when we know the number of iterations.

3. For each is used with array or STL.

What should be used in for loop i++ or ++i ?

Anyone can be used.

++i; is known to be faster than i++;.

Reason:

i++; is same as i=i+1; here i+1 is evaluated and the result is stored in a temporary variable and then assigned to i.

++i; here i+1 is not stored in a temporary variable, directly i is increased by 1

Factorial for large numbers

if the data type is taken long long then factorial will work upto 20!

if you want beyond 20! then take a string and write own procedure to calculate factorial.

Armstrong Number, why number is stored in m ?

actual number is taken in n.

While loop will stop when n=0

We have to compare sum with the number, so it is already stored in m.

Reverse a number ending with zero.

if a number is ending with 0 and you want 0 also then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';
cout<<reverse;
```

Display a number in words.

if a number is ending with 0 and you want to display in words then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';

for(int i=0;reverse[i]!='\0';i++)
{
	switch(reverse[i]-'0')
	{
		case 0: cout<<"Zero ";
			break;
		case 1: cout<<"One ";
			break;
		case 2: cout<<"Two ";
			break;
		.
		.
		.
	}
}
```

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

/*program for reversing a number


*/
int main()
{
      int n,r,rev=0;
      cout<<"enter the number";
      cin>>n;
      while(n>0)
      {
            r=n%10;
               r=n/10;
            rev=rev*10+r;
      }
      cout<<"reverse number is"<<rev;
      return 0;
}
```

while and for difference

while is used when you don't know how many time you have to repeat, so repeat WHILE condition is true.

for is used when you know FOR how many time you have to repeat.

When to use which loop ?

1. While and doWhile are similar, they are used when we don’t know number of times of repetitions

2. For is used when we know the number of iterations.

3. For each is used with array or STL.

What should be used in for loop i++ or ++i ?

Anyone can be used.

++i; is known to be faster than i++;.

Reason:

i++; is same as i=i+1; here i+1 is evaluated and the result is stored in a temporary variable and then assigned to i.

++i; here i+1 is not stored in a temporary variable, directly i is increased by 1

Factorial for large numbers

if the data type is taken long long then factorial will work upto 20!

if you want beyond 20! then take a string and write own procedure to calculate factorial.

Armstrong Number, why number is stored in m ?

actual number is taken in n.

While loop will stop when n=0

We have to compare sum with the number, so it is already stored in m.

Reverse a number ending with zero.

if a number is ending with 0 and you want 0 also then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';
cout<<reverse;
```

Display a number in words.

if a number is ending with 0 and you want to display in words then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';

for(int i=0;reverse[i]!='\0';i++)
{
	switch(reverse[i]-'0')
	{
		case 0: cout<<"Zero ";
			break;
		case 1: cout<<"One ";
			break;
		case 2: cout<<"Two ";
			break;
		.
		.
		.
	}
}
```

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

/*program for finding gcd of 2 number


*/
int main()
{
      int m,n;
      cout<<"enter two numbers"
      cin>>m>>n;
      while(m!=n)
      {
            if(m>n)
                  m=m-n;
            else if(n>m)
                  n=n-m;
      }
      cout<<"gcd of 2nos"<<m;
      return 0;
}
```

Program to find GCD

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

int main()
{
    int m,n;
    cout<<"Enter two no.s "<<endl;
    cin>>m>>n;

    while(m!=n)
    {
        if(m>n)
            m=m-n;
        else
            n=n-m;
    }

    cout<<"GCD is "<<m;

    return 0;
}
```

Program to Reverse a number

```cpp
#include <iostream>
using namespace std;

int main()
{
    int n, num, digit, rev = 0;

    cout << "Enter a positive number: ";
    cin >> num;

    n = num;

    do
    {
        digit = num % 10;
        rev = (rev * 10) + digit;
        num = num / 10;
    } while (num != 0);

    cout << " The reverse of the number is: " << rev << endl;

    if (n == rev)
        cout << " The number is a palindrome";
    else
        cout << " The number is not a palindrome";

    return 0;
}
```

while and for difference

while is used when you don't know how many time you have to repeat, so repeat WHILE condition is true.

for is used when you know FOR how many time you have to repeat.

When to use which loop ?

1. While and doWhile are similar, they are used when we don’t know number of times of repetitions

2. For is used when we know the number of iterations.

3. For each is used with array or STL.

What should be used in for loop i++ or ++i ?

Anyone can be used.

++i; is known to be faster than i++;.

Reason:

i++; is same as i=i+1; here i+1 is evaluated and the result is stored in a temporary variable and then assigned to i.

++i; here i+1 is not stored in a temporary variable, directly i is increased by 1

Factorial for large numbers

if the data type is taken long long then factorial will work upto 20!

if you want beyond 20! then take a string and write own procedure to calculate factorial.

Armstrong Number, why number is stored in m ?

actual number is taken in n.

While loop will stop when n=0

We have to compare sum with the number, so it is already stored in m.

Reverse a number ending with zero.

if a number is ending with 0 and you want 0 also then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';
cout<<reverse;
```

Display a number in words.

if a number is ending with 0 and you want to display in words then,

Example : n=2500

You have to store the reverse number in a string

```
char reverse[10];
int r,i=0;
while(n>0)
{
	r=n%10;
	reverse[i]=r+'0';  // '0' is added to make a digit as character.
	n=n/10;
}
reverse[i]='\0';

for(int i=0;reverse[i]!='\0';i++)
{
	switch(reverse[i]-'0')
	{
		case 0: cout<<"Zero ";
			break;
		case 1: cout<<"One ";
			break;
		case 2: cout<<"Two ";
			break;
		.
		.
		.
	}
}
```