Introduction to C++

C++ is Object Oridented programming language.IT is developed by Bjarne Stroutrup at AT& T Laorartories.C++ is a combination of the object oriented features of a language called Simula 67 with the elegance of C.C++ also called as extention of C with classes.

The First C++ Program

```
#include <iostream.h>

Void main()

{

Cout<<"welcome to C++";

}
```

Like C , C++ also is a collection of functions.Every program must have main function from where the excecution starts.

Every statement ends with semicolon except the few control statements and definations.

We can give single line comments using // and block comments using /* and */

The output operator used here is cout << "welcome to C++" ; it displays the string "welcome to c++" on screen.cout is a predefined object that represents the standard output stream in C++.A stream is flow of data.<< is a operator called as insertion operator ,it is not necessary to specify the data typeof the variable on its right.The insertion operator automatically assumes the data type of the variable.One advantage of this is that user-defined data types can be used with this operator which was not possible with printf() function in C.

The identifier cin is predefined object that corresponds to the standard input stream.The example will how the use of cin identifier.

```
#include <iostream.h>

void main()

{
char name[12];
cout<<"enter your name";
cin>>name;
cout<<"Welcome to C++"<<name;
}
```

The >> is extraction operator used with cin which works similar to scanf() function in C.>> will stop reading the input when the first white space is encountered.

The iostream.h header file contents the declaration for the identifier cout,cin and the operator <<, >>.This header file must be included at the beginning for the program that use input / output statements.

Another example of C++ program

```
#include <iostream>

void main()

{
std::cout<<"Welcome to C++"<<name;
}
```

Is same as

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

{
cout<<"Welcome to C++"<<name;
}
```

Above both the programs are same.If you have a working copy of iostream.h, it is probably the same as iostream except that everything in iostream is in the namespace std, while iostream.h generally preceded namespaces, and didn't use them.

using namespace std;

All the elements of the standard C++ library are declared within what is called a namespace, the namespace with the name std.

```
namespace std
{

}
```

So in order to access its functionality we declare with this expression that we will be using these entities. This line is very frequent in C++ programs that use the standard library.
In <iostream> the components are declared in namespace std whereas in <iostream.h> components are in global scope .

Benefits of OOP :-

1. Object-oriented programming is modular. The objects can be extended to include new attributes and behaviors. Objects can also be reused within an across applications. Main advantage of object oriented programming is modularity, extensibility, and reusability.Object-oriented programming provides improved software-development productivity over traditional procedure-based programming techniques.
2. In OOP, data can be made private to a class such that only member functions of the class can access the data. This principle of data hiding helps the programmer to build a secure program.
3. With the help of polymorphism, the same function or same operator can be used for different purposes. This helps to manage software complexity easily.
4. Large problems can be reduced to smaller and more manageable problems. It is easy to partition the work in a project based on objects.
5. It is possible to have multiple instances of an object to co-exist without any interference i.e. each object has its own separate member data and function.
6. Through the inheritance we can eliminate redundant code and extend the use of existing classes.
7. We can build programs from the standard working modules that communicate with one another rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.
8. It is possible to have multiple objects to coexist without any interface.
9. Object Oriented systems can be easily upgraded from small to large systems
10. Software complexity can be easily managed

Applications of OOP :- Main application areas of objected oriented programming are listed below : -

1. User interface design such as windows, menu etc.
2. Artificial Intelligence
3. CIM / CAD (Computer Integrated Manufacture / Computer Aided Design) system
4. Decision Support System
5. Real Time Systems
6. Simulation and Modeling
7. Object oriented databases
8. AI and Expert System
9. Neural Networks and parallel programming
10. Decision support and office automation system etc

Concepts of OOP (Object Oriented Programming) :-

- Object and Classes
- Data Abstraction and Encapsulation
- Dynamic Memory Manipulation
- Constructors and Destructors
- Inheritance
- Polymorphism
- Dynamic Binding
- Message Communication

Data Abstraction

Data abstraction is the ability to create user defined data types for modeling real worldobjects using built in data types and a set of permitted operators.The constructs that helps create such data type and implement data abstraction in C++ is the class.

Introduction to Classes and Objects

A class declaration in C++ is similar to structure declaration in C.However the main idea behind using classes is the binding of data along with its functionality.

Example : -If we want to declare a class for students to ask his/her name and rollno and to display it then a class will be written as shown below :

Syntax of declaration of class :

```
class classname
{
Access specifier : lsit of data members ;
Access specifers :  list of member functions;
}
```

Encapsulation

Encapsulation or data hiding is the mechanism that associates the code and the data that it manipulates into a single unit and keeps them safe from external interference and misuse.C++ implements the concept of encapsulation by providing access specifiers.Access Specifers control the visibility status of the members of a class.There are three access specifiers in C++

1. private – The members declared in private can only be accessed within class.
2. Public – The members declared in public can be accessible anywhere within the class or outside the class.
3. Protected – The members declared in protected can be accessed from the derived classes but cannot be accessed from anywhere in other classes or the main program.(It will be used in inheritance).

Example of class student :-

```
class student
{
private : int name,rollno;
public: void input()
{
cout<<"\n enter your name and rollno";
cin>>name>>rollno;
}
void display()
{
cout<<"\n student's rollno is <<rollno;
cout<<"\n students's name is <<name;
}
};
```

How to declare Object: -

Syntax : -

classname objectname1,objectname2,… ;

Example : -

If we are declaring an object of class student then

student s;

where s is a object (variable) of class student.This object can access all the members declared inside public access specifier of that class with a dot (.) operator.

Example :- s.input() or s.display()

Dynamic Memory Manipulation

C++ provide two special operators to perform memory management dynamically.

The new operator

The general fromat of the new operator is as follows :

datatype = new datatype [size in integer];

where the datatype is the return type the pointer to that type which will be returned and new is the new operator.the datatype is again the type for which memory is to be allocated and size is the number of items to be allocated (this is optional). So the statement is

example :- student *s;

s= new student;

new operator is similar to malloc () library function used for memory management.

The delete operator

The delete operator ensures the safe and efficient use of memory. This operator is used to free the block of memory allocated by the new operator. Although the memory allocated is returned to the system when the program terminates,it is safer to explicitly free memory using the delete operator.

Syntax of delete operator : -

Delete pointervariable;

Where the pointervariable is the pointer returned through the new operator.

Example : -

delete s;

delete operator is similar to free( ) library function used for memory management.

Note

If the new operator is used to explicitly allocate memory to a variable the delete operator must be used to destroy it. Though the variable goes out of scope when the program gets terminated,the memory area allocated by the new operator is not released to the system unless the delete operator explicitly released it.