

Module-1

Introduction to Object Oriented Programming: Computer programming background- C++ overview- First C++ Program -Basic C++ syntax, Object Oriented Programming: What is an object, Classes, methods and messages, abstraction and encapsulation, inheritance, abstract classes, polymorphism.

Textbook 1: Chapter 1(1.1 to 1.8)

Prerequisite:

The student should know the :

What is Procedure Oriented Programming (POP). Example

list of programming languages(High-Level) examples

Difference between an array and a structure.

what is a structure and why it is used.

What is an Object.

What is an Object?

An Object is a software entity that models something in the real world.

It has two main properties:

State: the object encapsulates information about itself - attributes or fields.

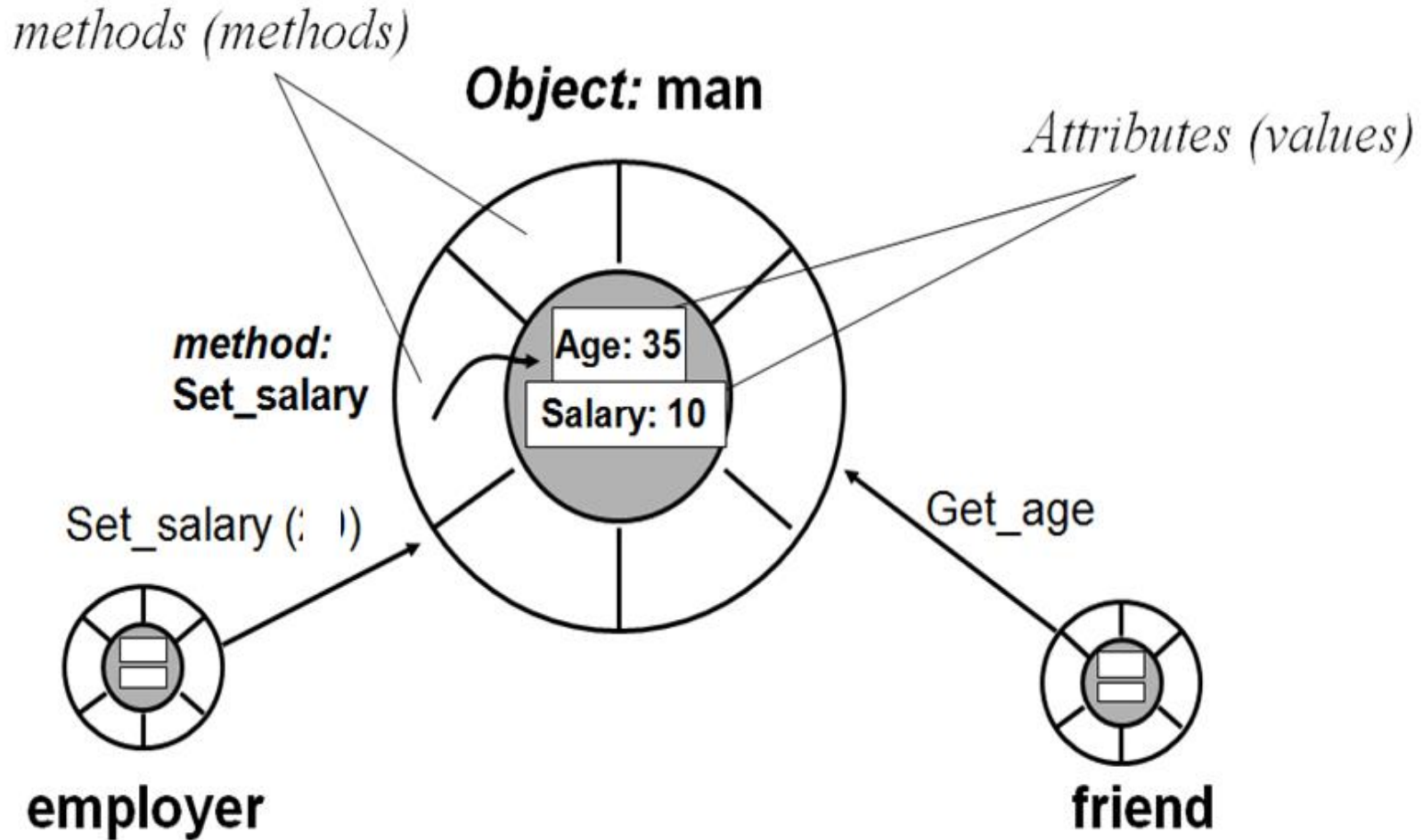
Behaviour: the object can do some things on behalf of other objects – methods

Example: In a banking system, a particular bank account is an example of an object.

Its state consists of attributes like: owner, account number etc.

Its behaviours consist of: deposit, withdraw, etc.

Objects



Programming Languages:

Programming languages allow programmers to code software.

The three major families of languages are:

- Machine languages
- Assembly languages
- High-Level languages (c ,c++,java etc)

A **high-level language** is a programming **language** such as C that enables a programmer to write programs.

Such **languages** are considered **high-level** because they are closer to human **languages** .

The syntax of HL languages is similar to English.

HL languages into two groups:

- Procedural -Oriented Programming Languages (**POP**)
- Object-Oriented Programming Languages (**OOP**)

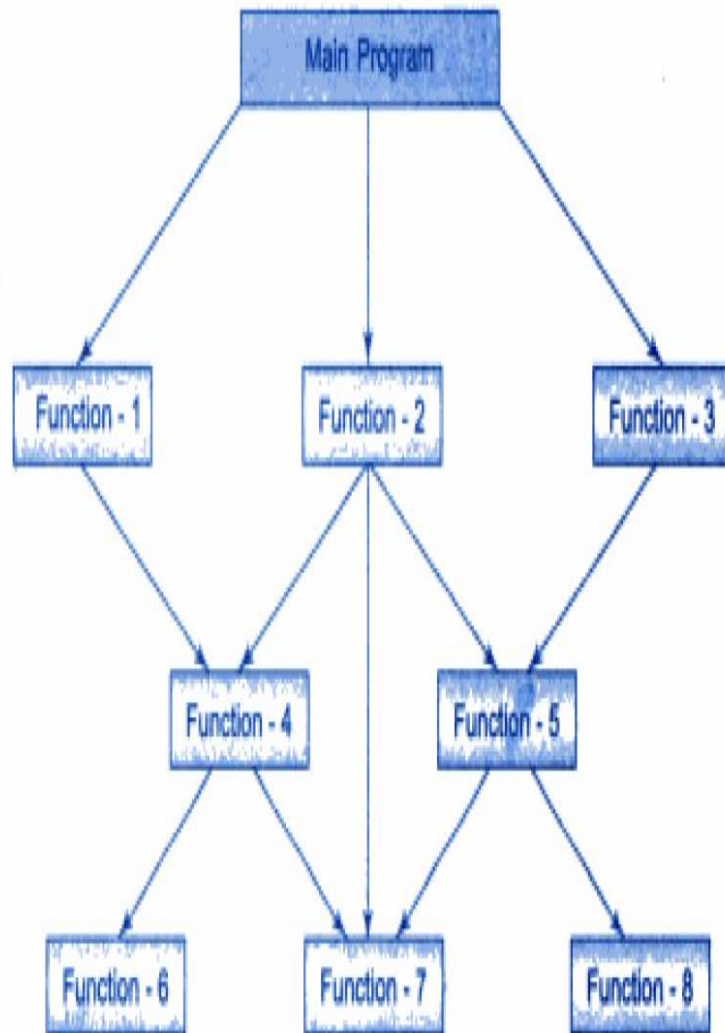
Basic concepts (features) of Object-Oriented Programming

1. Objects
2. Classes
3. Data abstraction
4. Data encapsulation
5. Inheritance
6. Polymorphism

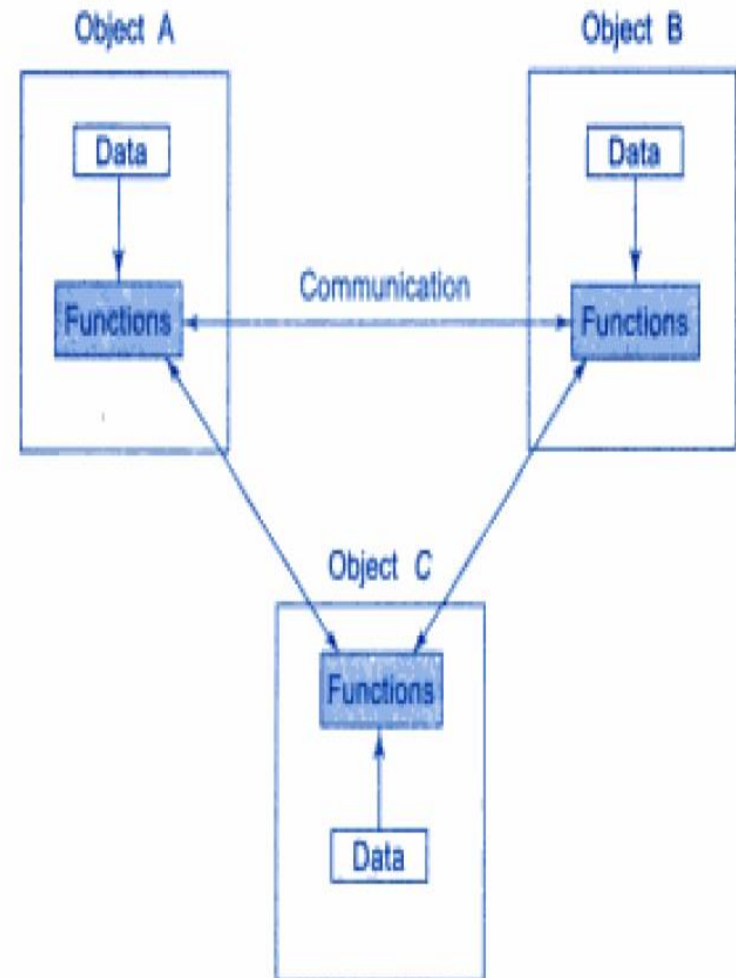
What is ? Why ?

Procedure Oriented Programming (POP)	Object Oriented Programming (OOP)
As the name implies, Procedure Oriented Programming contains step by step procedure to execute . Example: C, etc	As the name implies Object-oriented programming (OOP) is a programming language model organized around objects Examples include C++,Java, C#.NET, etc
the problem is viewed as a sequence of things to be done, such as, input taking, calculating and displaying.	objects
the problems get decomposed into small parts and then to solve each part one or more functions are used.	objects
Decompose the main problem in small parts called functions.	In OOP, program is divided into parts called objects .
Fig: next ppt	

Fig: Typical structure of procedure oriented programs



Typical structure of Object oriented programs



In POP, Importance is not given to data but to functions as well as sequence of actions to be done i.e. Emphasizes on functions.	In OOP, Importance is given to the data rather than procedures or functions because it works as a real world. Emphasizes on data.
POP follows Top Down approach.	OOP follows Bottom Up approach.
POP does not have any access specifier.	OOP has access specifiers named Public, Private, Protected, etc.
In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.
To add new data and function in POP is not so easy.	OOP provides an easy way to add new data and function.
In POP, Most function uses Global data for sharing that can be accessed freely from function to function in the system.	In OOP, data can not move easily from function to function , it can be kept public or private so we can control the access of data.
POP does not have any proper way for hiding data so it is less secure.	OOP provides Data Hiding so provides more security.
In POP, Overloading is not possible.	In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.
programming is step-by-step, in a really long program it becomes tough to back and follow up on the developments.	programming is Block by block or module by module , in a really long program it becomes tough to back and follow up on the developments.
Modification of a completed program is very difficult and it may affect the whole program.	Modifications are easy as objects stay independent to declare and define.

Disadvantages or limitations of Procedure Oriented Programming (POP) but advantages of Object Oriented Programming (OOP)

Or

Benefits of Object Oriented Programming over Procedure Oriented Programming :-

- 1) **No security for data.** If we declare a variable before main function then it can be accessed freely from any function present in the program.
OOP: using access specifiers , encapsulation
- 2) No better memory management. **OOP: memory management tech...**
- 3) Difficult to implement today's client requirements.
- 4) No structure or code reusability, **OOP: using inheritance. Reusability**
- 5) As length of application increases it causes slow performance.
- 6) It is difficult to create new data types. The ability to create the new data type of its own is called extensibility. Structured programming languages are not extensible.
OOP: using inheritance.
- 7) No proper way method for Exception handling.
- 8) We can build programs from the standard working **modules** that communicate with one another, rather than having to start writing the code from scratch which happens procedure oriented approach. This leads to saving of development time and higher productivity.
OOP: using Abstraction
- 9) Software complexity can be easily managed.

What are the characteristics of Object Oriented programming language

Or principal features of OOP are:

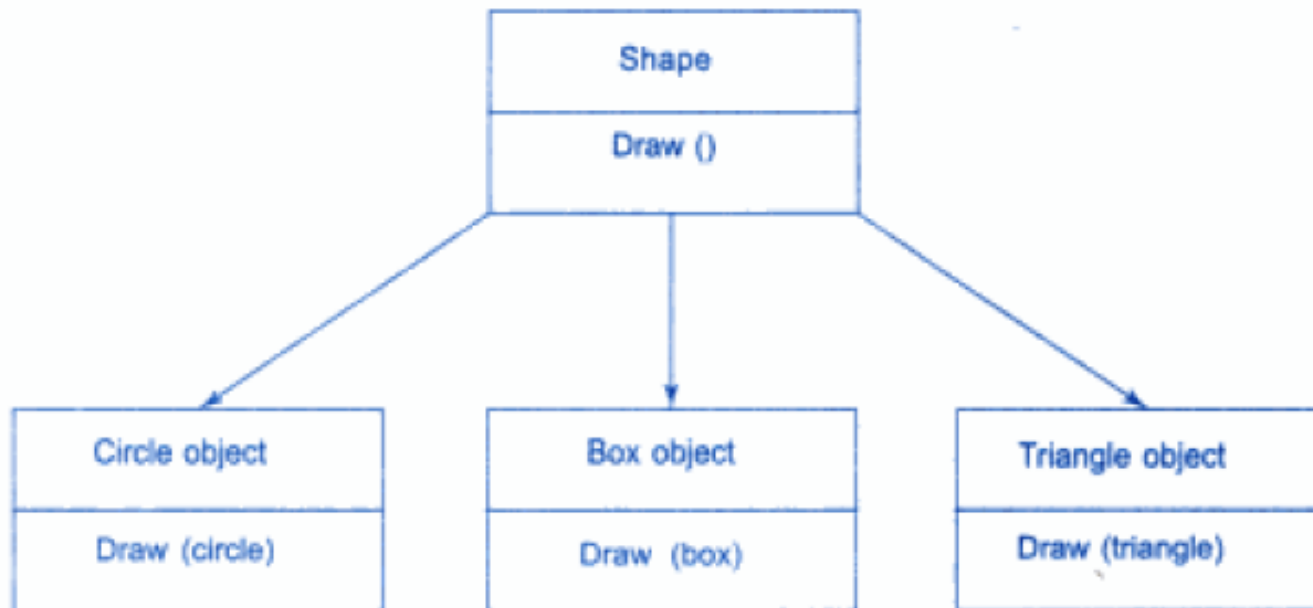
- 1) Object. The instances of a class which are used in real functionality
- 2) Class: These contain data and functions bundled together under a unit. class is a collection of similar objects. Basic building blocks OOP
- 3) Abstraction: The ability to represent data at a very conceptual level without any details.
- 4) Data Hiding and Encapsulation: Encapsulation is capturing data and keeping it safely and securely from outside interfaces.
- 5) Inheritance: This is the process by which a class can be derived from a base class with all features of base class and some of its own. This increases code reusability.



6) Polymorphism

Polymorphism: Poly means many. Morphism means forms.

This is the ability to exist in various forms. For example an operator can be overloaded so as to add two integer numbers and two floats.

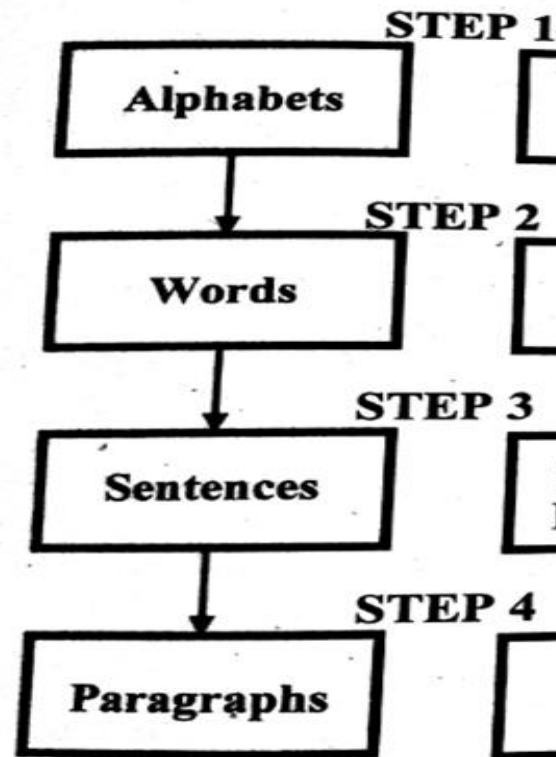


What is the difference between C and C++? Or comparison of C++ with C

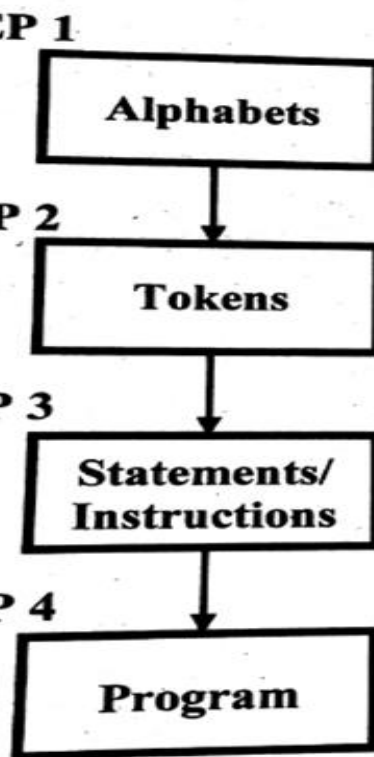
C	C++
C is a procedural programming language	C++ is a combination of both procedural and object oriented programming language; therefore C++ can be called a hybrid language.
does not support classes and objects	support classes and objects
C was developed by Dennis Ritchie between 1969 and 1973	C++ was developed by Bjarne Stroustrup in 1979
When compared to C++, C is a subset of C++.	C++ is a superset of C. C++ can run most of C code while C cannot run C++ code.
C does not support object oriented programming; therefore it has no support for polymorphism, encapsulation, and inheritance.	Being an object oriented programming language C++ supports polymorphism, encapsulation, and inheritance.
In C (because it is a procedural programming language), data and functions are separate and free entities.	In C++ (when it is used as object oriented programming language), data and functions are encapsulated together in form of an object. For creating objects class provides a blueprint of structure of the object.

In C, data are free entities and can be manipulated by outside code. This is because C does not support information hiding.	In C++, Encapsulation hides the data to ensure that data structures and operators are used as intended.
C does not support function and operator overloading.	C++ supports both function and operator overloading.
C does not have namespace feature.	C++ uses NAMESPACE which avoid name collisions.
C uses functions for input/output. For example scanf and printf.	C++ uses objects for input output. For example cin and cout.
C does not support reference variables.	C++ supports reference variables.
C has no support for virtual and friend functions.	C++ supports virtual and friend functions.
C provides malloc() and calloc() functions for dynamic memory allocation, and free() for memory de-allocation.	C++ provides new operator for memory allocation and delete operator for memory de-allocation.
C does not provide direct support for error handling (also called exception handling)	C++ provides support for exception handling. Exceptions are used for "hard" errors that make the code incorrect.

Steps to learn
English language



Steps to learn
C/C++ Language



Ex: Letters: A to Z, a to z
Digits: 0 to 9
Symbols: +, -, *, <, > etc.

Ex: Keywords: for, if, int, float,.....
Variables: sum, area, first, marks,.....
Constants: 10, 10.5, -5.9999, 'a',.....

Ex 1: area = length * breadth;
Ex 2: si = p * t * r / 100;

```
#include <stdio.h>
void main()
{
    printf(" Bangalore");
}
```

```
#include<iostream.h>
void main()
{
    cout<<"My First C++ Program";
}
```

**Fig. Steps to learn a
programming language**

The Structure of a C++ program :The general basic structure of a C++

[Comments /Documentation Section]

[Pre-processor Directives/***Header Files***/]

[Global Declaration Section]

void main()

{

Declaration Section

Executable part

}

<p>In C</p> <p>scanf() Function used for Input.</p> <p>printf() Function used for output.</p>	<p>In C++</p> <p>Cin>> Function used for Input.</p> <p>Cout<< Function used for output.</p>
<pre>#include<stdio.h> void main() { printf("My First C Program"); }</pre>	<pre>//TC #include<iostream.h> void main() { cout<<"My First C++ Program"; }</pre>
C++ in TC	C++ in Ubuntu and CodeBlocks
<pre>#include<iostream.h> void main() { cout<<"My First C++ Program"; }</pre>	<pre>//Ubuntu and CodeBlocks #include<iostream> using namespace std; int main() { cout<<"God"; }</pre>
<pre>gedit filename.c cc filename.c C in Ubuntu ./a.out</pre>	<pre>gedit filename.cpp g++ filename.cpp C++ in Ubuntu ./a.out</pre>

Function of iostream.h

The I/O library predefines a set of operations for handling reading and writing of build-in data types.

Predefined Stream Objects

cin (pronounced "see-in"), as istream class object tied to standard input. cin stands for console input.

cout (pronounced "see-out"), as ostream class object tied to standard output. cout stands for console output.

Console Output/input in C++

Cin: used for keyboard input.

Cout: used for screen output.

Since Cin and Cout are C++ objects, they are somewhat “Intelligent”.

- They do not require the usual format strings and conversion specifications
- They do automatically know what data types are involved.
- They do not need the address operator
- They do require the use of the stream **extraction** (>>) and **insertion** (<<) operators.

C++ Output Operator

The **output operator** ("<<") ("put to"), also called stream insertion operator is used to direct a value to standard output.

```
#include<iostream>
```

```
void main()
```

```
{
```

```
cout<<"Welcome to VVCE \n" ;
```

```
}
```

Output: **Welcome to VVCE**

C++ Input Operator

The **input operator** (">>") ("get from"), also known as stream extraction operator is used to read a value from standard input.

```
#include<iostream>
```

```
void main()
```

```
{
```

```
int val1, val2;
```

```
cin>>val1>>val2;
```

```
cout<<val1<<val2;
```

```
}
```

Output: **2 3**

```
/* C++ Program - Get Input from User */
```

```
#include<iostream.h>
```

```
void main()
```

```
{
```

```
    int num;
```

```
    cout<<"Enter a number : ";
```

```
    cin>>num;
```

```
    cout<<"You entered "<<num;
```

```
}
```

When the above C++ program is compile and executed, it will produce the following result:

```
Enter a number : 234
You entered 234
```

/* C sum of two numbers */

```
#include<stdio.h>
void main()
{
int a, b, sum;
printf("\nEnter two no: ");
scanf("%d %d", &a, &b);
sum = a + b;
printf("Sum : %d", sum);
}
```

/* C++ sum of two numbers */

```
#include<iostream>
using namespace std;
int main()
{
int a, b, sum;
cout<<"Enter the two number : ";
cin>>a>>b;
sum = a+ b;
cout<<"The sum of the two number is : "<<sum;
}
```