Session on C++ and OOP’s concept

Session held by –
Shivani Subhash Kagale

Topics

1.) Introduction to C++

2.) History

3.) Difference between C & C++

4.) Features

5.) OOP’s concept

6.) Advantages of OOP’s

What is C++…?

- C++ is a general purpose, case sensitive, free form programming language that supports object-oriented, procedural and generic programming.

- C++ is a middle-level language as it encapsulates both high & low level features.

C++ history

C++ programming language was developed in 1980 by Bjarne Stroustup at bell laboratories of AT & T (American Telephone & Telegraph), located in U.S.A.

Bjarne Stroustup is known as the founder of C++ language.

It was develop for adding a feature of OOP(Object Oriented Programming) in C without significantly changing the C components.

C++ programming is relative (called a “superset”) of C it means any valid C program is also a valid C++ program.

Difference between C & C++

C

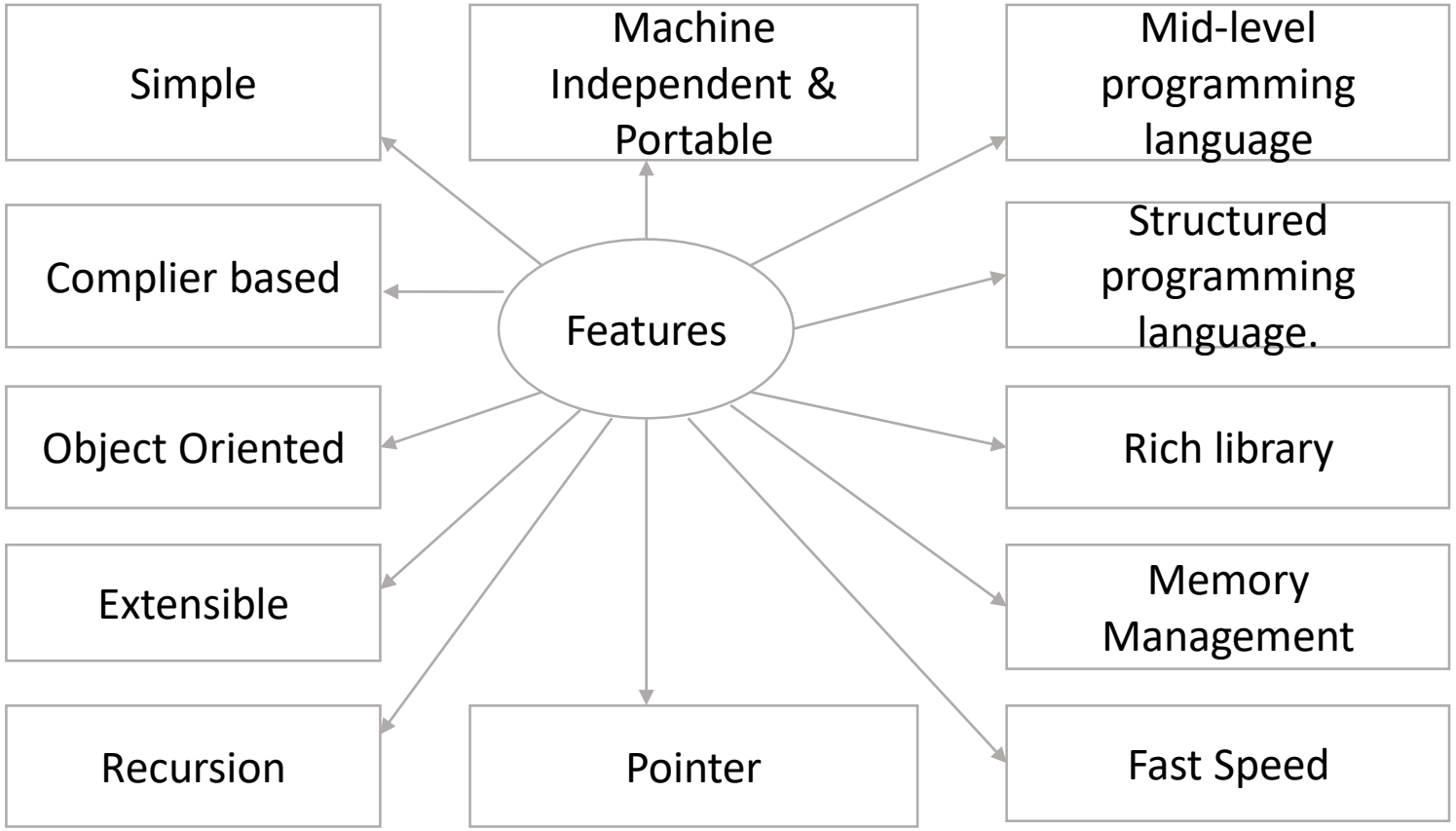
- C follows the procedure style programming.
- Data is less secured in C.
- C follows the top & down approach.
- C does not support function overloading.
- In C, you cant use function in structure
- In C, scanf() & printf() are mainly used for input/output.
- Operator overloading not possible.
- C does not support the inheritance.

C++

- C++ is multiple-paradigm. It supports both procedural & object oriented.
- In C++ you can use modifiers for class members & member functions to make it inaccessible for outside users.
- C++ follows the bottom up approach.
- C++ supports function overloading.
- In C++, you can use function in structure.
- In C++, mainly uses stream cin & cout to perform input & output operations.
- Operator overloading is possible.
- C++ supports inheritance.

C++ features

C++ is object oriented programming language. It provide a lot of features that are given below.



1.) Simple –

C++ is a simple in the sense that it provides structural approach (to break the problems into parts), rich of library functions, data types, etc.

2.) Machine Independent or Portable –

Unlike assembly language, C program can be executed in many machine with little bit or no change. But it is not platform independent.

3.) Mid- level programming language –

C++ also used to do low level programming. It is used to develop system application such as kernel, drivers, etc. It also supports the features of high level language. That is why it is known as mid level language.

4.) Structural programming language –

C++ is a structural programming language in the sense that we can break the program into parts using functions. So it is easy to understand & modify.

5.) Rich library –

C++ provides a lot of inbuilt functions that makes the development fast.

6.) Memory management –

It supports the features of dynamic memory allocation. In C++ language, we can free the allocated memory at anytime by calling the free() function.

7.) Speed –

The compilation & execution time of C++ language is fast.

8.) Pointer –

C++ provides the features of pointer. We can directly interact with memory by using the pointer. We can use the pointer for memory, structure, function, array.

9.) Recursion –

In C++, we can call the function. It provides code reusability for every function.

10.) Extensible –

C++ language is extensible because it can easily adopt new features.

11.) Object Oriented –

C++ is object oriented programming language, OOP's makes development & maintenance easier where as procedure-oriented programming language it is not easy to manage if code grows as project size grows.

12.) Complier based –

C++ is a complier based programming language, it means without compilation no C++ program can be executed. First we need to compile our program using complier & then we can execute our program

OOP's concept – The major purpose of C++ programming language

Object oriented programming is a paradigm that provides many concept such as inheritance, data binding.

The programming paradigm where everything is represented as an object is known as truly object oriented programming language.

OOP's (Object Oriented Programming system) –

Object means a real word entity such as chair, table, etc. Object oriented programming is a methodology or paradigm to design a program using classes & object.

It simplifies development & maintenance by providing some concepts –

- ✓ Object
- ✓ Class
- ✓ Inheritance
- ✓ Polymorphism
- ✓ Abstraction
- ✓ Encapsulation

Class

It is user defined datatype, which holds its own data member and member functions, which can be accessed & used by creating an instance of the class. Class is blue print for an object. Collection of object is called class. It is logical entity.

- A class is user-defined data-type which has data member & member function.
- Data member are the data variables & member functions are the function used to manipulate these variables together these data member & functions define the properties and behaviour of the object in a class.

A class is defined in C++ using keyword class followed by the name of class. The body of class is defined inside the curly brackets & terminated by a semicolon.

```
class Classname
{
        Access Specifier;     //can be private ,public or protected
        Data member;          //variable to be used
        Member function;      //method to access data member
};      //class name ends with semicolon
```

Object

Any entity that has state behaviour is known as an object, for e.g. chair, pen, table, car, etc.
It can be physical or logical. It is an instance of class. When a class is defined, no memory is allocated but when it is instantiated (i.e. is an object is created) memory is allocated.

Inheritance

When one object acquires all the properties & behaviour of parent object i.e., known as inheritance.
It provides code reusability. It is used to achieve runtime polymorphism.

Polymorphism

When one task is performed by different ways i.e. known as polymorphism, for e.g. to make marketing of product, to get to know something.
In C++, we use function overloading and over riding to achieve polymorphism.

Abstraction

Hiding internal details & showing functionality is known as abstraction, for e.g. phone call, we don't know the internal process. In C++, we use abstract class & interface to achieve abstraction.

Encapsulation

Binding or wrapping code & data together into a single unit is known as encapsulation for e.g. capsule, it is wrapped with different medicine.

Advantages of OOP's over procedure oriented programming language

1.) OOP's makes development & maintenace easier where as in procedure-oriented programming language it is not easy to manage if code grows as project size grows.

2.) OOP's provide data hiding whereas in procedure-oriented programming language a global data can be accessed from anywhere

3.)OOP's provide ability to simulate real world event more effectively, we can provide the solution of real world problem if we are using the object oriented programming language.