OOP's in C++

Agenda

- What is OOP’s Concept
- Access Modifiers
- Constructors
- Encapsulation
- Abstraction
- Inheritance
- Polymorphism

What is OOP’s

OOP's Concept

- It stands for Object-oriented programming System.
- It is a type of programming paradigm that uses concept of classes and objects.
- It deals with Real-World Entity.
- Four Important pillars of OOP's are-

  Encapsulation
  Abstraction
  Inheritance
  Polymorphism

Class

- User-defined datatype which is having its own member function.
- Accessible by building instance by its own class.
- It is a blueprint for an object.
- Ex- Creating a class of human, humans are having eyes, nose and their name. These are properties and having functions such as sleep, walk, talk.

Object

- It is a real world entity having attributes and behavior.
- It is an instance of class.
- **Ex**- Desktop, Camera

Access Modifiers

Access Modifiers

- It is also known as **Access Specifiers**.
- It specify the accessibility of members which are declared in the class.
- It is of three types:

  Public- Members declared under class are accessible throughout the program.

  Private- Not accessible by function or method outside the class.

  Protected – accessible by derived classes.

Constructors

Constructors

- It is a special type of method which is used to initialize an object of a class.
- Constructor name must be same as class name.
- It does not have any return type.
- It must be public.

- **Types of Constructor:**

  Default Constructor- Constructor which is automatically generated by compiler.

  Parameterized Constructor- arguments are passed

Principals of Object-oriented Programming

Principles of OOP's

There are four principles of Object-oriented Programming System-

- **Encapsulation**
- **Abstraction**
- **Inheritance**
- **Polymorphism**

Encapsulation

- Bundling or tying data and its methods together that operate on that data so that they are grouped together within a class.
- Encapsulation has been done with the purpose of preventing anything or anyone outside of our class to be able to directly access our data and to interact with it and to modify it.
- It is used in hiding data.
- Setter and getter method are used in order to access the data while keeping the members private in class.

Abstraction

Abstraction means hiding complexity while showing the functionality.

Implementing a contract which is actually an abstract class. So function inside a abstract class is mandatory to implement in order to execute a contract.

Example - Withdrawl of money from ATM Machine but did not know the internal working.

Concept:

```
class Abstractstudent{
 virtual void scholarship()=0;
   };
 class student:Abstractstudent{
  }
  void scholarship(){
  }
  };
```

Inheritance

It is one of the most important principle of Object-oriented programming system in which derived class acquires the properties and behavior from a base class.

Parent/base/super class- It is a class whose properties are acquired by derived class.

Child/derived/sub class- It is a class which acquire properties from different class.

Syntax:

```
class sub_classname : base_classname
{

}
```

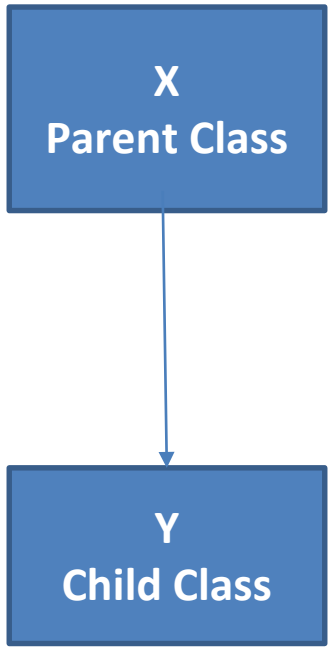
Types of Inheritance

There are five types of Inheritance-

- **Single Inheritance**
- **Multiple Inheritance**
- **Hierarchical Inheritance**
- **Multilevel Inheritance**
- **Hybrid Inheritance**

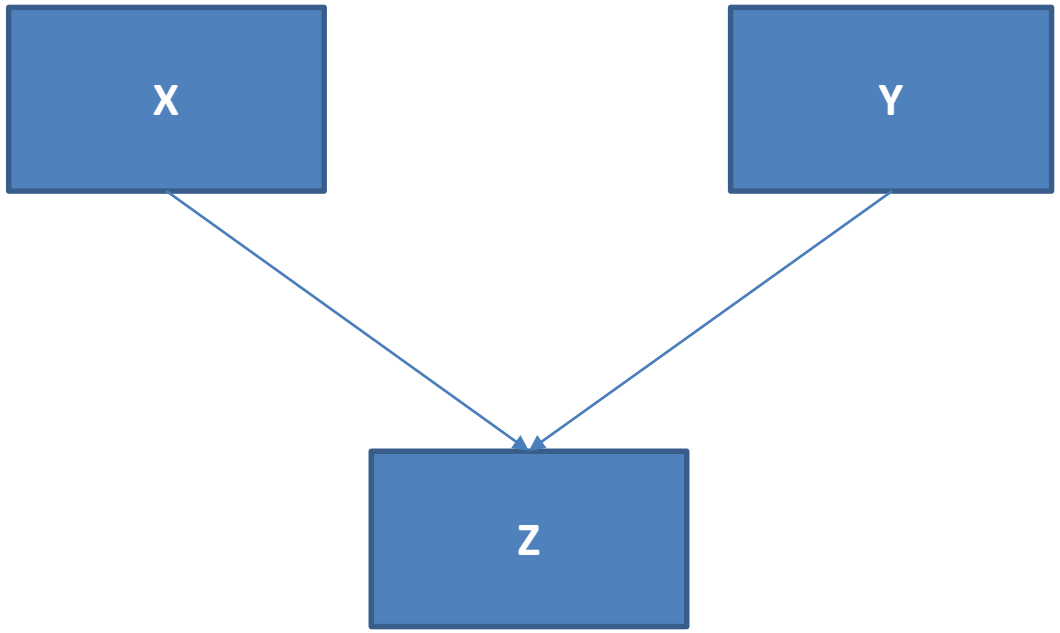
Single Inheritance

Child/sub/derived class is inherited by only one parent/base/super class.



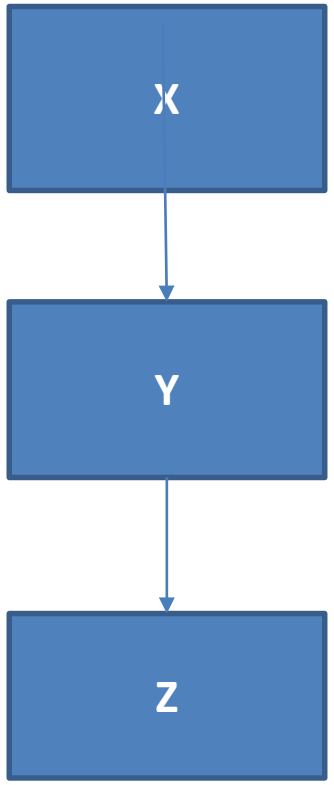
Multiple Inheritance

Single derived class is inherited from two or more base class.



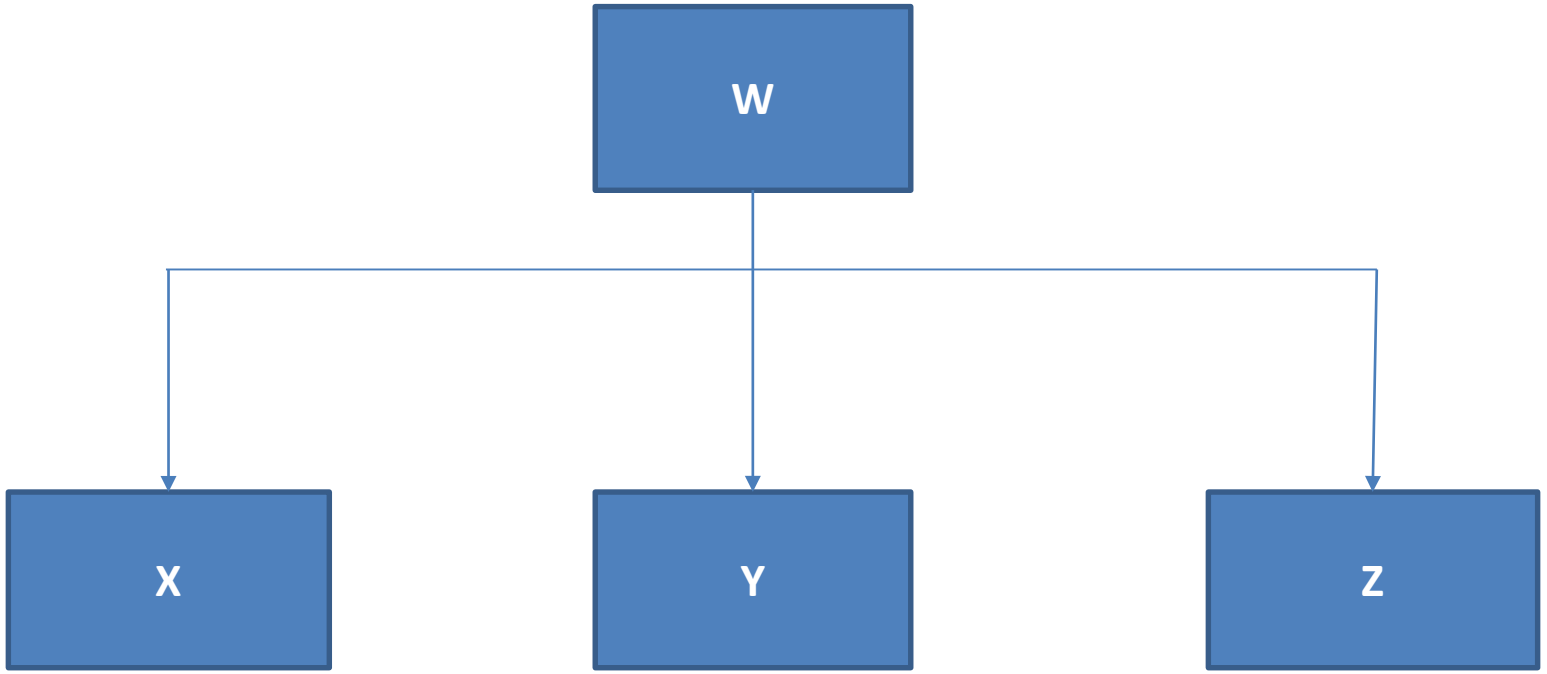
Multilevel Inheritance

Sub class is constructed from another sub class.



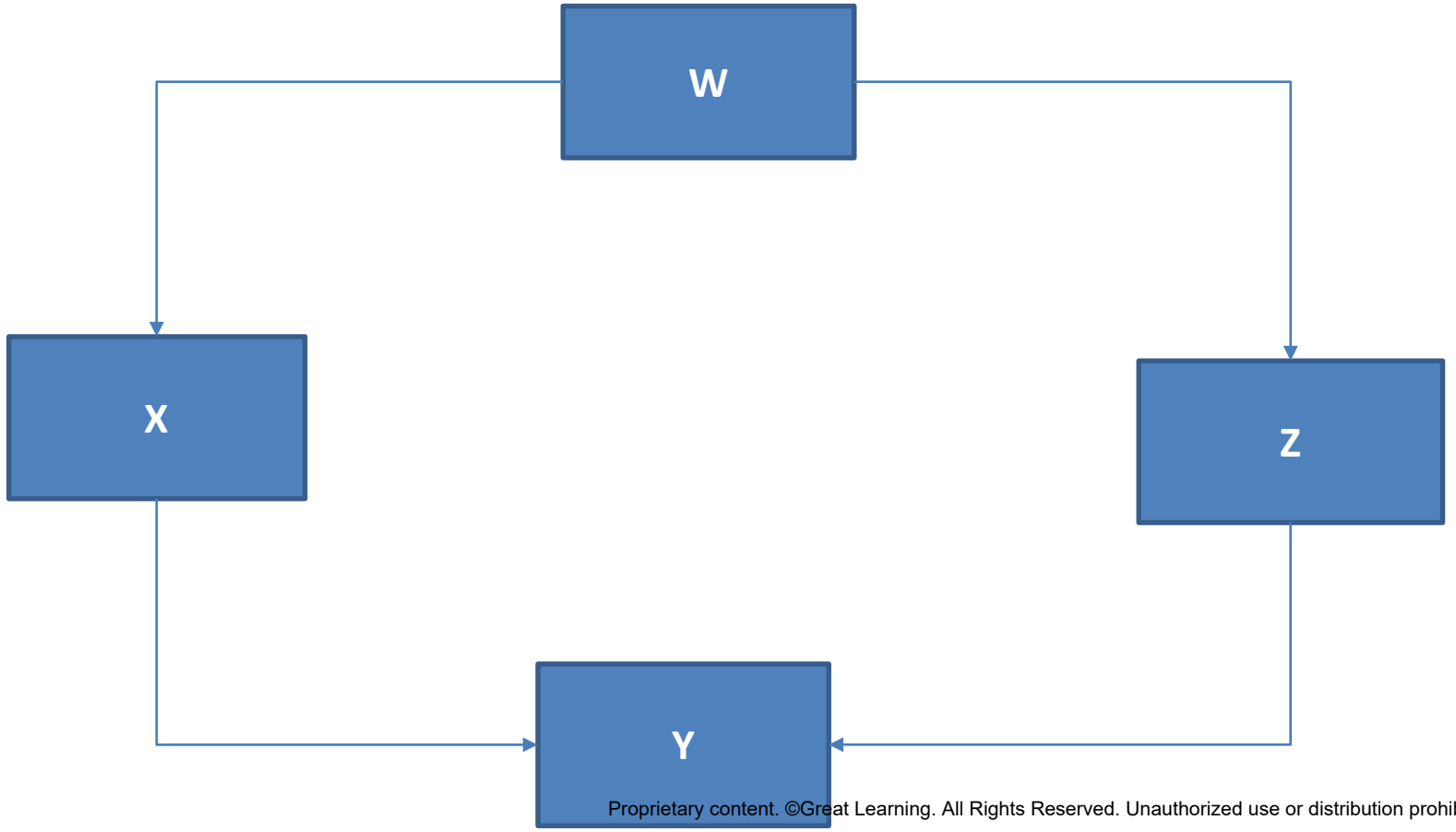
Hierarchical Inheritance

More than one derived/sub class is constructed by a one super/base class.



Hybrid Inheritance

Association of two or more types of inheritance.



Polymorphism

Polymorphism is an ability of an entity to have various forms.

It permits to execute a single activity in distinct ways.

Example- A boy at the same time is student as well as son of another person.

It is of two types –

Compile time Polymorphism

Runtime Polymorphism

Types of Polymorphism

Compile time polymorphism-

It is known as early binding or static binding.

Function is invoked at compilation time.

Accomplished by function overloading and operator overloading.

Function Overloading- Function having same name with different parameters

Operator Overloading- specify extra tasks to operators without converting its real meaning.

Types of Polymorphism

Run time polymorphism-

It is known as Dynamic or late binding.

Function is invoked at execution time.

Accomplished by virtual functions and pointers.

Function overriding is used.

Function Overriding- Two methods or more having same name with different parameter.

Summary

- Classes and Objects
- Access Modifiers
- Constructor
- Four Principals of OOP’s

Thank You