Tech-tutorials 2k21

learn subjects easily

Master Core Java in 30 days

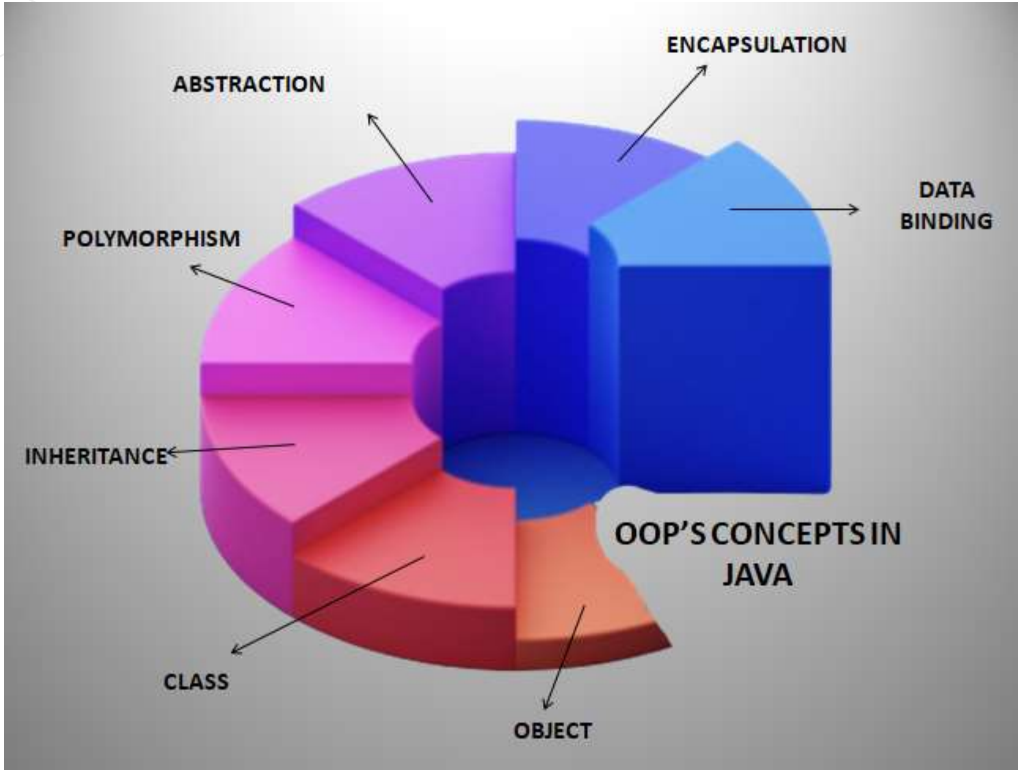
Day-6

Object Oriented Programming Concepts

Core Java

@5:30 PM Today

Object Oriented Concepts in Java



Object

- Any entity that has state and behaviour is known as an object.
- An Object can be defined as an instance of a class. An object contains an address and takes up some space in memory. Objects can communicate without knowing the details of each other's data or code
- The state of an object is stored in fields (variables), while methods (functions) display the object's behaviour. Objects are created at runtime from templates, which are also known as classes.

Example

Identity
Name of the Employee

State/Attributes
Eno
Ename

Behaviour (Functions)
setemp()
display(0

State: It is represented by attributes of an object. It also reflects the properties of an object.
Behaviour: It is represented by methods of an object. It also reflects the response of an object with other objects.
Identity: It gives a unique name to an object and enables one object to interact with other objects.

Class

Class is a collection of data members and member functions.

The variables declared inside of a class is said to be data members of a class

The functions declared inside of a class said to be member functions

Syntax:

```
<access specifier> class class_name
{
 data members
 member functions
}
```

Example

```
Public class Employee
{
int eno;
String ename;
Public:
Void setemp(int e,String n);
Void display();
}
```

Inheritance

The process of an object can acquire the properties of another class is said to be inheritance

Or

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object.

Class: A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.

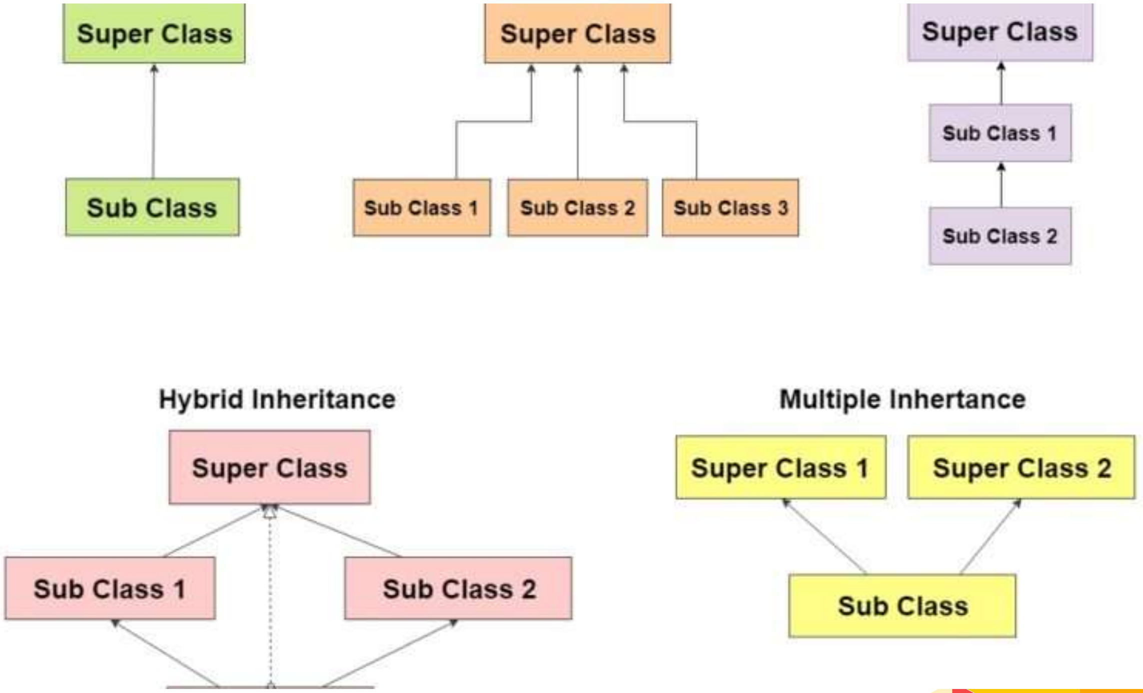
Sub Class/Child Class: Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.

Super Class/Parent Class: Super class is the class from where a subclass inherits the features. It is also called a base class or a parent class.

Reusability: As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

Types of Inheritance

```
class Subclass name (dept)
extends
Superclass-
name  (Employee)
{
  //methods and fields
}
```



Polymorphism

Polymorphism is the ability of an object to perform different actions (or, exhibit different behaviours) based on the context.

Types of polymorphisms

Compile time Polymorphism (Method Overloading)
Run time polymorphism (Method Over riding)

Encapsulation

Encapsulation in Java is a mechanism for wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes and can be accessed only through the methods of their current class. Therefore, it is also known as **data hiding**.

To achieve encapsulation in Java −

Declare the variables of a class as private.

Provide public setter and getter methods to modify and view the variables values.

Abstraction

Abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the user.

In other words, the user will have the information on what the object does instead of how it does it In Java, abstraction is achieved using Abstract classes and interfaces.

data binding

Binding means an association of method call to the method definition. There are two types of Binding: **Static and Dynamic Binding in Java**.

If the compiler maps the method at compile-time, it is Static Binding or early binding. And, if the method is resolved at runtime, it is Dynamic Binding or late binding.

Static Binding or Early Binding in Java refers to a process where the compiler determines the type of object and resolves the method during the compile-time. Generally, the compiler binds the overloaded methods using static binding.

data binding

When the compiler resolves the method call binding during the execution of the program, such a process is known as Dynamic or Late Binding in Java. We also call Dynamic binding as Late Binding because binding takes place during the actual execution of the program.

The best example of Dynamic binding is the Method Overriding where both the Parent class and the derived classes have the same method. And, therefore the type of the object determines which method is going to be executed.

The type of object is determined during the execution of the program, therefore it is called dynamic binding.

Tech-tutorials 2k21

Thank You

Please Subscribe, Like and Share
My videos