**Object Oriented Programming Fundamentals**

1. What is the main difference between a class and an object?

Answer:

Object Oriented Programming is a methodology of designing programs with classes and Objects.

A class is a template or prototype of an object. Properties and behaviour of objects are decided in the class.

Whereas an object is a real world entity which has its own properties and behaviour.

In other words, an object is an instance of a class.

1. What is Encapsulation? Explain with a used case

Answer:

Encapsulation is the mechanism of wrapping up of data and code together as a single unit. This is achieved by declaring attributes as private and access them using public getter and setter methods. Encapsulation leads to data hiding.

Employee

Class Name

Private Attributes

-EmpName

+get EmpName()

+set EmpName()

Public Methods

Emp1

Emp1.set EmpName(“David”)

Emp1.get EmpName()

Employee Object

1. What is Polymorphism? Explain with a used case

Answer:

Polymorphism is a concept in Object Oriented Programing that allows an object to take multiple forms. There are 2 types of Polymorphism.

* Compile time polymorphism or Static polymorphism which is resolved at Compile time. Method Overloading, Constructor Overloading are examples
* Runtime Polymorphism or Dynamic Polymorphism that is resolved at Runtime. Method Overriding is an example of Runtime polymorphism.

Customer

+ searchItems()

+ purchaseItems()

Online Customer

+ searchItems()

+ serachItems(item)

+purchaseItems()

1. Explain Overriding & Overloading and its advantages

Answer:

**Overloading** is an example of compile time polymorphism. Methods will have same name but number and type of arguments can be different and also return type can be different

Example: Add()

Int Add(a,b)

Add(a,b,c)

Advantages:

* Increases readability of code
* Cleanliness of Code
* Programmers have the freedom to call the same method with different type of data.

**Overriding** is an example of runtime polymorphism. Overridden methods have same name, argument list and return type. But they may be at different class. These methods are called by the reference variable of superclass.

Advantages:

* Helps in writing generic code based on parent class
* Multiple implementation of same method

1. What is Inheritance and different types of inheritance? Explain with a used case

Answer:

Inheritance is a property in Object Oriented Programming that allows the child class to acquire the properties from parent/super class.

Types of Inheritance:

Single Inheritance: There will be one parent class and one child class. The child class will get properties and behaviour from single parent.

Class A

Parent Class

Class B

Child Class

Multilevel Inheritance: There will be a Parent class, child class and grandchild class. Grandchild class will have properties of Child class as well as Parent class.

Class A

Parent Class

Class B

Child Class

Class C

Grandchild

Hierarchical Inheritance: There will be one parent class and more than one child class.

Class A

Parent

Class D

Class C

Class B

Child

Multiple Inheritance: There is will more than one parent class and only one child class. Thus child class will acquire properties from both parent. In Java Multiple inheritance is implemented by Interface.

Parent

Class B

Class A

Class C

Child

1. What is an abstract class?

Answer:

* Abstract class is used to achieve abstraction which is hiding the implementation details from user.
* Abstract class is a class that contain one or more abstract methods and non-abstract methods. Abstract method is not implemented in the same, it will be implemented in another class which is extended from this abstract class.
* Abstract class must be declared with the keyword Abstract
* Abstract class cannot be instantiated.

1. What is an interface and how multiple inheritance is achieved with this?

Answer:

Interface is another mechanism to achieve abstraction in Object Oriented Programming. Interface is the keyword used to identify an Interface class. In interface class methods will be initialised but it won’t be implemented. These methods will be implemented in another class.

Multiple inheritance can be achieved through interface because a child class can implement the methods from more than one class through interface.

Interface Class1

Interface Class2

Method2

Method1

Class3

Implement Method1

Implement Method2

Multiple Inheritance

1. What are the access modifiers?

Answer:

Access modifier is a concept in Object Oriented Programming to set accessibility of class, methods and attributes. Types of access modifiers are public, private, protected and default.

* Public: It can be accessed from everywhere within the class, outside the class and outside package.
* Private: Private methods can be accessed only within the class
* Protected: Can be accessed only through child class.
* Default: Can be accessed within the package

1. What are the various types of constructors?

Answer:

A constructor is like a method (without return type) which initiates an object. Constructor will have same name as the class name. Types of constructors are

* Default Constructor: If programmer didn’t write any constructor code in program the compiler will insert a constructor at the time of compilation.
* Zero Argument Constructor: Constructor with no parameters/arguments in it.
* Parameterized: Constructor with at least one parameter.

1. What is ‘this’ pointer?

Answer:

This keyword is used inside a class in Java to indicate the current instance. A class can have multiple object. ‘this’ pointer identifies the instance of an object.

1. What is static and dynamic Binding?

Answer:

Static binding happens at compile time and dynamic binding happens at the run time. Method overloading is an example of static binding whereas method overloading is an example of dynamic binding.

1. How many instances can be created for an abstract class and why?

Answer:

We cannot create any instance or object for an abstract class. Because abstract class is like a template. It won’t have implementation details for abstract methods. So we need to extend this class to another class which implement the methods. Instances can be created for that extended class.

1. Which OOPS concept is used as a reuse mechanism and explain with a use case

Answer:

Inheritance concept in OOP is used in reuse mechanism. Because the child class can acquire the properties and methods from its super class. Only unique methods for the class need to be mentioned, others can be reused from the parent

Customer

Parent Class

+ searchItems()

+ purchaseItems()

Online Customer

Child Class

+ onlinepayment()

Object of Online Customer

Search Items

Purchase items

Online payment

1. Please identify one practical scenario for each pillar of OOPs.

Answer:

Four pillars of OOP are:

* Encapsulation: Machines like television, Fan, washing machine shows the nature of encapsulation and abstraction.
* Inheritance: Car can be considered as an example for Inheritance. Every model of car will have some basic features like steering, 4 wheels, brake etc. But each model will have their own extra features.
* Abstraction: Mobile Phone can be considered as an example for Abstraction. Users will dial, send message, watch videos, play games but they are not showing how mobile actually works. Implementation details are hide from users.
* Polymorphism: Through phone we can send text message, voice message and video message.