**OPP--- Encapsulation**

1. **What is Java and what are the four fundamental concepts of Java?**

Java is a object oriented programming language in which programs are **considered** as a collection of objects. Each object is nothing but an instance of a class. It has four fundamental concepts :a) Abstraction b) Inheritance c. Polymorphism d) Encapsulation.

1. It is the features or characteristics of OPP language **. Encapsulation in Java** is a process of wrapping code and data together into a single unit.

How - Declare the variables of a class as private. Provide public setter and getter methods to modify and view or get the variables values.

**Why encapsulation is used in Java?**

It is used to hide the implementation details from users. If a data member is private it means it can only be accessed within the same class. No outside class can access private data member (variable) of other class. That's **why encapsulation** is known as data hiding. It used to hide the internal representation, or state, of an object from the outside.

**Real Life Example -**In my project I used Encapsulation for hiding details from the third parties. To hide the actual implementation I define the variable as private so that from outside class no one can access it . To access this variable I used getter and setter method to set the value and get the value from the other class.

1. **What is the primary benefit of encapsulation**? It improves the securities of data, We can change the implementation of all these getter and setter any time. So it will give us a very good amount of the flexibility and maintenance of the code as well as reusability.
2. **What are the features of encapsulation?**

a)It combining the data of our application and its manipulation at one place. B)It allows the state of an object to be accessed and modified through behavior.c) It allows to change one part of code without affecting other part of code.d) It allows to control who can access what. & It also helps to write immutable class in Java which is a good choice in multi-threading environment.

### Mention some important points about encapsulation in Java. Encapsulation in Java is achieved using access modifiers private, protected and public. Factory pattern' and 'singleton pattern' in Java makes good use of encapsulation.

# **Realtime Examples & Advantages and Why it is use ?**

Suppose you have an account in the bank. If your balance variable is declared as a public variable in the bank software, your account balance will be known as public, In this case, anyone can know your account balance. So, Nobody like this.   
So, they declare balance variable as private for making your account safe, so that anyone cannot see your account balance. The person who has to see his account balance, he will have to access private members only through methods defined inside that class and this method will ask your account holder name or user Id, and password for authentication. **Thus, We can achieve security by utilizing the concept of data hiding. This is called Encapsulation.**

1. **What should you encapsulate in code?** Anything which can be changed or which is more likely to be changed in near future is candidate of encapsulation. This also helps to write more specific and cohesive code. For instance object creation code, code which can be improved in future like sorting and searching logic.

**OPP--- -Inheritance**

1. **What is inheritance?** Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. Inheritance represents the IS-A relationship, also known a parent-child relationship.
2. **What is multiple inheritance?** If a child class inherits the property from multiple classes is known as multiple inheritance.
3. **Does Java support multiple Inheritance? And Why?** Java doesn't support multiple Inheritance using classes. A class can extend only one class but it can implement multiple interfaces. Java supports only single inheritance , that is, you can only inherit one class at a time.
4. **Why we need to use Inheritance?** A) For Method overriding (used for Runtime Polymorphism) B) It’s main uses are to enable polymorphism and to be able to reuse of code for different classes by putting it in a common super classes. C)The biggest advantage of inheritance is that the code that is already present in base class need not be rewritten in the child class. D) To implement parent child relationship. a)super b) this c) extent d) extends
5. **What are the types of inheritance?**

* 1).Multiple inheritance( java doesn't support multiple inheritance). 2.Multilevel inheritance.

1. **How Inheritance can be implemented in java?** Inheritance can be implemented in JAVA using below two keywords:  1)**extends-**extends is used for developing inheritance between two classes and two interfaces. 2**) implements**- implements keyword is used to developed inheritance between interface and class.
2. **What is multilevel inheritance? How do you restrict a member of a class from inheriting to it’s sub classes.?  What happens if super class and sub class having same field name?**

* Getting the properties from one class object to another class object level wise with different priorities.
* By declaring that member as a private. Because, private members are not inherited to sub classes.

### Super class field will be hidden in the sub class. You can access hidden super class field in sub class using super keyword.

### How do you implement multiple inheritance in java?

### Ans: Using interfaces java can support multiple inheritance concept in java. in java can not extend more than one classes, but a class can implement more than one interfaces.  Program:

interface A{ } **class C** **implements** A,B{}

}

interface B{ }

### Are constructors inherited? Can a subclass call the parent's class constructor? When?

### No We cannot inherit a constructor.

### No a subclass cannot inherit the constructors of its superclass. Constructors are special function members of a class in that they are not inherited by the subclass. Constructors are used to give a valid state for an object at creation.

### You know that all classes in java are inherited from java.lang.Object class. Are interfaces also inherited from Object class.? No, only classes in java are inherited from Object class. Interfaces in java are not inherited from Object class. But, classes which implement interfaces are inherited from Object class.

1. **Can we reduce the visibility of the inherited or overridden method ?** --No.

### 12) Does a class inherit the constructor of its super class? No.

### Which of the following is tightly bound ? Inheritance or Composition ? Inheritance.

**In case of Composition** - Composing object holds a reference to composing classes and hence relationship is loosely **bound**. and Its the relationship between objects.

**In case of Inheritance** - Single class can only **inherit** 1 Class and Derived object carries the base class definition in itself and hence its **tightly bound**. Inheritance IS-A relationship between classes

### 13 ) Difference Between this() and super() ?

### Keyword “this ” is a reference variable in java that refers to the current object. It can be used—

### to refer instance variable

### to invoke or instantiate current class constructor and

### it can be passed as an argument in the method call.

### The “Super ‘’ keyword refers to super class (parent) objects. It is used

### To call super class methods

### To access the super class constructor

### To eliminate the confusion between super class and sub class that have methods with same name.

### 

### 14)Can an interface be extended by another interface in Java ?

An interface can be extended by another interface in Java.  The code for the same would be like as shown below:

### //this interface extends from the Body interface: public interface FourLegs extends Body { public void walkWithFourLegs( );}

### OPP--- ****-Polymorphism****

### ****1).What is Polymorphism?** Polymorphism is a mechanism where a parent class reference variable can take many forms(it can refer object from different classes). More than one function with same name, with different working.**

**2)Polymorphism two types---** 2 ways of 1)Method/Function Overloading and 2) Method /Function Overriding

a) **Compile time /Static polymorphism**. –is called as early binding polymorphism. Example-1) Function overloading and 2) Operator Overloading. Example-Method Overloading And Constructor overloading. Memory will be allocated at compile time.

b)**Runtime /Dynamic Polymorphism** -is also called late binding polymorphism. Example-Virtual Functions, Example-Method Overriding. Memory will be allocated at run time. Runtime time polymorphism is done using inheritance and interface.

Method/Function Overloading-Two or more function can have same name but different parameters.

**3) Which object oriented Concept is achieved by using overloading and overriding? --** Polymorphism.

### 4) How is Polymorphism supported in Java?

### Java has excellent support of polymorphism in terms of 'inheritance', 'method overloading' and 'method overriding'. 'Method overriding' allows Java to invoke method based on a particular object at run-time instead of declared type while coding.

### 5)Example of Method Overriding -

### a) Method overriding -- Overriding means having two methods with the same method name and parameters (i.e., method signature). One of the methods is in the parent class and the other is in the child class. Overriding allows a child class to provide a specific implementation of a method that is already provided its parent class.

### Example --There are two type of Testing: one which is drove by human is called Manual testing and other is driving by automation tools is called Automation testing that’s overriding when same name but different way of working and different parameters .

### 6 -a) Method overloading— Overloading occurs when two or more methods in one class have the same method name but different parameters.

### a)Method Overloading- To verify the Test case scenario like LoginHomePage. We create positive and negative test cases (valid and invalid ) to know the result that is successfully expected or not. For that purpose we create more than one methods in the same class and same method name with different parameters based on the test cases.

### b) Method overloading: The payment option on any ecommerce website has several options like netbanking, COD, credit card, etc. That means, a payment method is overloaded several times to perform single payment function in various ways.

### 

### 7) How to overload a contractor method? Yes. Normal method overloading and Contractor method overloading are same except the following—in normal method -method should be called by using object reference but in case of Constructor method – method should be called without object refernce and it has no return type.

OPP--- **-Polymorphism**

**8) What is method overloading and method overriding?**

**Method overloading**

Is a process that allows a class to have two or more methods having same name with different parameter declarations. These methods are called overloaded method.

* Points to be remember about method overloading’s---Same method names,Parameter lists are different ,Inside the same class. The methods must have different signature. Only **method** name and Parameters are **defined** as **method signature**. THE **METHOD SIGNATURE INCLUDES** THE **RETURN TYPE.** It may or may not need inheritance in Method Overloading.

**9)Method overriding**

Declaring a method in subclass which is already present in superclass is know as Method overriding.(Same Name with same parameters)

Rules for M. overriding – Name, Signature type, parameter must be same. If a method cannot be inherited, then it can not be overridden. A method declared as final or static can’t be overridden.

Constructor can’t be overridden. It cannot be done within a class, for this we required derived class and base class. It always requires inheritance in Method Overriding. In Method Overriding, sub class have the same method with same name and exactly the same number and type of parameters and same return type as a super class.. It is a run time polymorphism.

**Can you override a private or static method in Java?**

You cannot override a private or static method in Java. If you create a similar method with same return type and same method arguments in child class then it will hide the super class method; this is known as method hiding.

Similarly, you cannot override a private method in sub class because it’s not accessible there. What you can do is create another private method with the same name in the child class.

**10)Why do we need –Method Overriding -**Code reuse,One interface, multiple implement,Run time polymorphism.

**11)Can we override static method---** Ans –No, Static method can’t be overridden. Because:----

* Static method is bound to class, on the other hand method is bound to object.
* Static Keywords relates to class and methods relates to object.

### 12)What is the difference between 'Overloading' and 'Overriding'?

|  |  |
| --- | --- |
| **Method overloading** | **Method overriding** |
| It occurs in a single class. And within the same class | It occurs at least in two class where is a super class and a subclass. |
| Parameter list must be different | exactly the same type of parameters |
| No inheritance ,inheritance is not involved . | ,Inheritance is involved . |
| Return type may or may not be same. | Return type must be same. |
| One method does not hide another method. | Child method hides the parent method. |

### 13) What is constructor overloading?

### Constructor overloading in java allows having more than one constructor inside one class.

### Constructor overloading is very much similar to the method overloading. But the basic differences is that 1) Constructor can’t return any value that’s why they don’t have any return type. 2) the Name of constructor must be same to the class name.3) Constructor will be called automatically when we create anew object.

### Example of Method Overriding -

### a) In case of Abstract Class- Abstract class can be extended by the another class and when extended by another class, all unimplemented method should be implemented in the another class with showing @ override. If the abstract class has the child by extending the child class. The child responsibility to implement on the unimplemented method and define the method body, (i.e)that means write the logic and the selenium over there. This are @override. Here we are overriding these method. Same method name, same parameter and return type and same signature are same, that is method overriding i.e in the parent class.

### 6 -a) Method overloading—Like I create a method. name is “doLoging”. So doLoging is there. First Method is for loging with UN and PWD. Public Homepage doLogin( String UN, String PWD){}. if I want to achieve method overloading . So we can do some negative testing with method overloading. So I have to do one thing that “doLoging” only without any UN and PWD. I will not pass anything that means (i.e) blank parameter instead of UN and PWD. So this is same method name one is zero parameter and same method name and two parameter or same method name and with UN and blank PWD that is called method overloading.

### OPP--- -Abstraction

### 1)What is Abstraction?

### Abstraction is the process of hiding the implementation details and showing only the functionality to the user. When we will not create object of any class, then we will use abstract class. Example -public abstract class A{ }

### 2)Rules of Abstraction ---1) If we use abstract word in a method, then the class(who hold this method) should be abstract. No body in abstract method that means no curley breze , no syso. 2) we can’t create object of abstract class but we can create abstract class constractor. 3) abstract method should be written with abstract keyword and without method body.4) In abstract class – method can be abstract and Non abstract. 5) Abstract class can be extended by the another class and when extended by another class, all unimplemented method should be implemented in the another class with showing @ override.

**3)How to achieve Abstraction? T**here are two ways to achieve abstraction in java.

* Abstraction Class( 0% to 100%) i.e you may abstract 100% or few or half or 0%,Abstracts depend on you.
* Interface ( Achieve 100% abstraction)

4)Real life **example** of **Abstraction Like**  ATM Machine----All are performing operations on the ATM machine like cash withdrawal, money transfer, retrieve mini-statement…etc. but we can't know internal details about ATM.

Note: Data **abstraction** can be used to provide security for the data from the unauthorized methods.

**Note:**In Java language data abstraction can achieve using class.

## Example of Abstraction

**class** Customer

{

**int** account\_no;

**float** balance\_Amt;

String name;

**int** age;

String address;

**void** balance\_inquiry()

{

/\* to perform balance inquiry only account number

is required that means remaining properties

are hidden for balance inquiry method \*/

}

**void** fund\_Transfer()

{

/\* To transfer the fund account number and

balance is required and remaining properties

are hidden for fund transfer method \*/

}

4)**What is Abstract Method ? When is an abstract method used?**

A) Abstract method is a method that is declared as abstract and It does not have implementation.it will be unimplemented .There is not variable, Is known as Abstract Method. And Abstract void demo(); No body Part. B) A method declared without a body(no implementation) within an abstract class is an abstract method. If we want a class to contain a particular method but we want the actual implementation of that method to be determined by the child classes, then we can declare the method in the parent class as an abstract.

### 5)What is an abstract class ? ****When we need abstract class?**** Can we directly call constructor of abstract class?

a) An abstract class is a class with collections of implemented and unimplemented method, which cannot be instantiated that means can not create an object of that class. It doesn't provide complete implementation and it enforce abstraction.

**b)The purpose of an abstract** class is to provide a common definition of a base class that multiple derived classes can share. And When we have to keep a method as unimplemented in a class and when we will not create object of any class then we will use abstract class.

**c) No-Abstract class’s contractor can’t call directly because we can’t create object of abstract class.**in that case we can call abstract class constructor by using “super” keyword from the subclass constructor. i.e we have to create constructor of subclass with super keyword (instead of “this” keyword).

### 6) Differentiate an Encapsulation and an Abstraction.

### Encapsulatio- Solves the problem in the implementation level. Encapsulation means hiding the code and data into single unit to protect data from the outside world. Inner layout- used in terms of implementation.

### Abstraction -Solves the problem in the design level. Used for hiding unwanted data and giving relevant result. Outer layout- used in terms of design.

### 7) Differentiate an Interface and an Abstract class.

An abstract class contain abstract method(without body) and Non abstract method. It doesn’t supports multiple inheritance. it can have static, non static, final and Nonfinal variable.

It has partial abstractor . Example-abstract class Animal{ abstract void eat()}

In case of abstract class,only abstract method should be implemented or override by the another class at mandatory way. But non abstract method of abstract class can be overridden, this is not mandatory.

**On the other hand**, Interface contain only abstract methods. It supports multiple inheritance . It can only have static and final variable. It fully abstract. Example- interface Animal{ void eat();}

In case of inheritance, we use implements keyword. In case of interface, all abstract method s/be implemented mandatory override by the another class.

Whereas an interface has all the public members, an abstract class contains only class members like private, protected and so on.

**Q9): Can an abstract class be declared final? What is use of a abstract variable? Can a abstract class be defined without any abstract methods? What is an abstract method? Can an abstract class be final?**

* No.
* Variables can't be declared as abstract.
* Yes it's possible. This is basically to avoid instance creation of the   class.
* An abstract method is a method which is unimplemented.
* An abstract class must not be declared as final.

### 8 ) How to define an abstract class ?

A class containing abstract method is called an abstract class. An Abstract class cannot be instantiated.  
**Example of Abstract class :**

abstract class testAbstractClass  
{  
protected String myString ;  
public String getMyString()  
{  
return myString ;  
}  
public abstract string anyAbstractFunction(); }

**10).What does it mean that a method or class is abstract?**  
 An abstract class cannot be instantiated. Abstract   methods may only be included in abstract classes.   However, an abstract class is not required to have any   abstract methods,   
though most of them do. Each subclass   of an abstract class must override the abstract methods   of its super classes or it also should be declared  abstract.

**Q11):When a class must be abstract class? When we need abstract class?**  
 An abstract class is a class with collections of implemented and unimplemented method, which cannot be instantiated (instantiated means we can not create object for abstract class)

* Even there is only one unimplemented method in a class then that class is known as Abstract class.
* When we have to keep a method as unimplemented in a class and when we will not create object of any class then we will use abstract class.
* we have to write that method as abstract method, then the whole class will be Abstract class.

abstract class can be extended by child class. We can not create object of abstract class but we can create constructor of abstract class. In abstraction, when unimplemented methods implemented in child class, overriding concept is happened. Abstract keyword should be used in abstract method and abstract class.