

**UNIT II – INHERITANCE AND INTERFACE**

**Part A – Question Bank**

**1. What is finalize () method?**

The finalize() method is called just before an object is garbage collected. It is used to dispose system resources, perform clean-up activities and minimize memory leaks.

**Syntax:**

```
protected void finalize()           // finalize() is called just once on an object
{
    .....
}
```

**2. What is meant by Inheritance?**

Inheritance is the mechanism in java by which one class is allow to inherit the features (fields and methods) of another class. It is process of deriving a new class from an existing class.

**Syntax:**

```
class Subclass-name extends Superclass-name
{
    //methods and fields
}
```

**3. List out the types of inheritance.**

- Single inheritance
- Multilevel inheritance
- Multiple inheritance
- Hybrid inheritance
- Hierarchical inheritance

**4. how do you implement multiple inheritance in java?**

Java does not allow multiple inheritance:

- To reduce the complexity and simplify the language
- To avoid the ambiguity caused by multiple inheritance

It can be implemented using Interfaces.

**5. Define super class and subclass.**

The process of deriving a new class from an existing class is inheritance. A class that is inherited is called a *superclass* and the class that does the inheriting is called a *subclass*.

## 6. state the use of keyword super.

- It can be used to refer immediate parent class instance variable when both parent and child class have member with same name
- It can be used to invoke immediate parent class method when child class has overridden that method.
- `super()` can be used to invoke immediate parent class constructor.

## 7. What is the use of Inheritance and what are its advantages?

Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.

## 8. Write short notes on Object class.

The Object class is the parent class of all the classes in java by default (directly or indirectly). The `java.lang.Object` class is the root of the class hierarchy. Some of the Object class are Boolean, Math, Number, String etc

## 9. Define abstract class?

Abstract classes are classes from which instances are usually not created. It is basically used to contain common characteristics of its derived classes. Abstract classes generally act as super classes. Methods can also be declared as abstract. This implies that non-abstract classes must implement these methods.

## 10. When to use abstract Methods & abstract class?

Abstract methods are usually declared where two or more subclasses are expected to do a similar thing in different ways through different implementations. These subclasses extend the same Abstract class and provide different implementations for the abstract methods.

Abstract classes are used to define generic types of behaviors at the top of an object-oriented programming class hierarchy, and use its subclasses to provide implementation details of the abstract class.

## 11. List out the rules in defining abstract classes.

- An abstract class may have concrete (complete) methods.
- An abstract class may or may not have an abstract method. But if any class has one or more abstract methods, it must be compulsorily labeled abstract.
- Abstract classes can have Constructors, Member variables and Normal methods.
- Abstract classes are never instantiated.
- For design purpose, a class can be declared abstract even if it does not contain any abstract methods.
- Reference of an abstract class can point to objects of its sub-classes thereby achieving run-time polymorphism Ex: `Shape obj = new Rectangle();`
- A class derived from the abstract class must implement all those methods that are declared as abstract in the parent class.
- If a child does not implement all the abstract methods of abstract parent class, then the child class must need to be declared abstract as well.

## 12. Define final class.

When a class is declared with *final* keyword, it is called a final class. A final class cannot be extended (inherited).

## 13. What are the uses of a final class?

- To prevent inheritance, as final classes cannot be extended. For example, all Wrapper Classes like Integer, Float etc. are final classes. We cannot extend them.
- To create an immutable class like the predefined String class. We cannot make a class immutable without making it final.

## 14. What are the benefits of final keyword in Java?

- Final** keyword improves performance. Not just JVM can cache final variable but also application can cache frequently use final variables.
- Final variables are safe to share in multi-threading environment without additional synchronization overhead.
- Final keyword allows JVM to an optimized method, variable or class.

## 15. Is final method inherited?

Yes, final method is inherited but we cannot override it.

### *For Example:*

```
class Bike{ final
    void run()
    {
    System.out.println("running...");
    }
}
public class Honda2 extends Bike
{
    public static void main(String args[]){
        new Honda2().run();
    }
}
```

## 16. What is blank or uninitialized final variable?

A final variable that is not initialized at the time of declaration is known as blank final variable.

If we want to create a variable that is initialized at the time of creating object and once initialized may not be changed, it is useful.

It can be initialized only in constructor.

**17. Define static blank final variable.**

A static final variable that is not initialized at the time of declaration is known as static blank final variable. It can be initialized only in static block.

**Example of static blank final variable**

```
public class A{
    static final int data;//static blank final variable
    static{ data=50;}
    public static void main(String args[]){
        System.out.println(A.data);
    }
}
```

**18. What is meant by interface?**

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types.

**19. What's the difference between an interface and an abstract class?**

An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, we have to inherit our class from it and Java does not allow multiple inheritance. On the other hand, we can implement multiple interfaces in your class.

**20. What are the uses of interface?**

- Interfaces are used to implement abstraction.
- Since java does not support multiple inheritance in case of class, it can be achieved by using interface.
- It is also used to achieve loose coupling.

**21. Define nested interface.**

An interface which is declared inside another interface or class is called nested interface. They are also known as inner interface. Since nested interface cannot be accessed directly, the main purpose of using them is to resolve the namespace by grouping related interfaces (or related interface and class) together.

**22. How will you find out the length of a string in java? Give an example?**

The length ( ) method is used to number of characters is string. For example,

```
String str="Hello";
System.out.println("Length of string is "+str.length( ));
```

**23. you are planning to do an indexed search in a list of objects. Which of the two java collections should you use: ArrayList or LinkedList?**

ArrayList

**24. how would you make a copy of an entire java object with its state?**

Have this class implement Cloneable interface and call its method clone().

**25. What is the basic difference between string and stringbuffer object? –**

String is an immutable object. StringBuffer is a mutable object.

**26. What is inner class?**

If the methods of the inner class can only be accessed via the instance of the inner class, then it is called inner class.

**27. What's the difference between an interface and an abstract class?**

An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.

**28. What would you use to compare two string variables - the operator == or the method equals()?**

I'd use the method equals() to compare the values of the Strings and the == to check if two variables point at the same instance of a String object.

**29. can an inner class declared inside of method access local variables of this method?**

It's possible if these variables are final.

**30. What's the main difference between a vector and an arraylist ?**

Java Vector class is internally synchronized and ArrayList is not.

**PART -B**

1. Explain overriding methods and final methods in Java.(8) (CS1261 MAY /JUNE 2016)
2. Create a Java class Shape with constructor to initialize the one parameter "dimension". Now create three sub classes of Shape with following methods: (16)  
(IT2301 APR/MAY-2015)
  - a. "Circle" with methods to calculate the area and circumference of the circle with di- mension as radius
  - b. "Square" with methods to calculate the area and length of diagonal of the square with dimension as length of one side.
  - c. "Sphere" with methods to calculate the volume and surface area of the sphere with dimension as radius of the sphere. Write appropriate main method to create object of each class and test every method.
3. Explain with example how multiple inheritance is achieved in Java. (16)  
(IT2301 APR/MAY-2015)
4. What are interfaces? Explain with an example how multiple inheritance is implemented using interfaces. (16) (CS2311 NOV/DEC-2014)

5. Develop a Library interface which has drawbook(), returnbook() (with fine), checkstatus() and reservebook() methods. All the methods are tagged with public in the following ways: (16) (IT2301 APR/MAY-2015)
  - a. Using draw book() - get the required book based on title
  - b. Using checkstatus – user book returned date details
  - c. Using with fine() – whether failed to return the book within a time period charge –  
Rs.5/day
  - d. Using reserve book() – block or reserve particular book for their account.
6. Explain about inheritance in Java. (16) (IT2301 NOV/DEC-2012) (CS1361 NOV/DEC- 2014)
7. Explain the interfaces in detail with suitable example. (CS2311 MAY/JUNE-2014)

SVCEET