**Core Java Concept Questions**

**OOPS:**

1. **What is oops?**

Ans: Oops is concept,oops is nothing but object oriented programming language.

1. **What is Encapsulation?**

Ans: Encapsulation is a mechanism binding “attributes and methods” together inside class and object creation,object creation through reference.

1. **What is Inheritance?**

Ans: Re-usability of existence functionality from super class to subclass.

1. **Is Java will support multiple Inheritance through classes?**

Ans: No.It will support through interfaces.

1. **What is polymorphism?**

Ans: Single entity shows multiple forms(technics).

1. **What is Overloading?**

Ans: It is compile time;methods are same,parameters should be different ,return type may or may not be same,it will happen same class or different class.

1. **What is Overriding?**

Ans: Method names are same,parameters are also same,return type also same ,it will happen two different class through inheritance.

1. **Difference Between overloading and overriding?**

Ans: **Overloading:** It is compile time;methods are same,parameters should be different ,return type may or may not be same,it will happen same class or different class.

**Overriding:**Method names are same,parameters are also same,return type also same ,it will happen two different class through inheritance.

1. **What is Abstraction?**

Ans: Hiding unnecessary details and functionalities ,showing only relevant data.

1. **What is object?**

Ans: Object is a instance of class.It is a real entity.

1. **What is class?**

Ans:class is a blue print of object.

**Constructor:**

1. **What is Constructor?**

Ans: Constructor is a special type,it is mainly used for initializing,constructor does not have any return type not even void also.

1. **How can we create constructor?**

Ans: Constructors are created while object creation.If we do not explicitly give a constructor then the JVM will create the default constructor internally called default constructor.

1. **How can we access constructor?**

Ans: We can Access constructors while creation of objects.

1. **How many ways we can create constructor?**

Ans: Two types:

1. default constructor,
2. Parameterized constructor.
3. **Can we declare constructor as void?**

Ans: No,if we declare as a void the compiler will consider as a method not a constructor.

1. **What is this keyword in constructor?**

Ans: This is keyword always pointing into the instance variable.

1. **What is super keyword in constructor?**

Ans: Super keyword always used to call super class constructor.

1. **What is copy constructor?**

Ans:By Copy Constructor object reference is copied & calls the constructor.

**Static**

1. **What is static?**

Ans:Static is a keyword.We can declare static as Methods & Variables.

1. **How can we declare the static keyword?**

Ans:We can access static variables & static methods directly through Class-name. method name and Class-name.variable name.

1. **Can static method access static methods and variables?**

Ans:Static methods can access static methods and static variables.

1. **Can static method access non-static methods and variables?**

Ans:Static methods can not access non-static methods and non-static variables.

1. **Can non-static method access static methods and variables?**

Ans:Non-Static methods can access static methods and static variables.

1. **What is static block?**

Ans: Whenever class is loaded into JVM at that time static block is initialized.

**Final**

1. **What is final?**

Ans: Final is a Keyword. We can declare final as Variables, Methods & Class. Final variables can not be modify. Final methods can not be Override. Final Class can not be extends (Inherited).

1. **How can declare final?**

Ans: We can declare final as Variables, Methods & Class.

1. **Can we modify the variables?**

Ans: No final variables cannot be modified.

1. **Can we override methods?**

Ans: No final methods cannot be override.

1. **Can we inherit classes?**

Ans: No final methods cannot be inherited.

1. **Can we initialize final variables while calling constructor?**

Ans: Yes,final variables also,we can initialize while constructor calling class variables are private and finally ,also attributes private.

**String**

1. **What is string?**

Ans: String is Final Class. Strings are immutable. String class having methods. All string class methods are non-synchronized.

1. **Why strings are immutable?**

Ans: Once we declare any string object, it is constant. If we are trying to modify an existing string it will create another memory location, the existing object is eligible for garbage Collection.

1. **Which package is string belongs to?**

Ans: String belongs to java.lang package.

1. **What are key features of strings in java?**

Ans: Immutable:once a string object is created,it cannot be changed.

Any modification to string creates a new string object.

1. **What is string pool in java?**

Ans**:** Java optimizes memory usage by storing strings in a special area of memory known as the “string pool”.

1. **How many ways we can create string?**

Ans: By using string literals and new key word.

1. **Is string is synchronize or non synchronize?**

Ans: non synchronize.

1. **What is string buffer?**

Ans: String buffer is Final Class. String buffer are mutable. String buffer class having methods. All string buffer class methods are synchronized.It is not recommended to use in development but still in API.

1. **What is synchronize and non synchronize?**

Ans: Synchronized- Tasks are performed in a single way like one by one.

Non-synchronized- All tasks are performed at a time randomly.

1. **What are methods in string ?**

Ans: equals(),toLowerCase(),toUpperCase(),Concat.

1. **What are methods in stringbuffer?**

Ans: append(),delete(),reverse(),replace(),length(),capacity().

**12.What is string Builder?**

Ans: String Builder is Final Class. String builder are mutable. String builder class having methods. All string builder class methods are non-synchronized.

1. **Difference between string,string buffer and string builder?**

Ans: String class: String is Final Class. Strings are immutable. String class having methods. All string class methods are non-synchronized.

String buffer :String buffer is Final Class. String buffer are mutable. String buffer class having methods. All string buffer class methods are synchronized.It is not recommended to use in development but still in API.

String builder:String Builder is Final Class. String builder are mutable. String builder class having methods. All string builder class methods are non-synchronized.

1. **What is the jdk version of string builder?**

Ans: jdk1.5.

1. **When we can do string as a immutable class?**

Ans: We can declare class as final. The class is declared as final so that it cannot be sub classed.

1. **What is thread safe?**

Ans: Synchronized(thread safe).

1. **What is the difference between == and equals()?**

Ans: == is used for compare the content of string.(object address)

Equals() comparing the value(content).

1. **What are the methods of string builder?**

Ans: append(),delete(),length(),replace(),reverse().

1. **Is string ,stringbuffer and string builder are final class?**

Ans: Yes, all are final classes.

**Interfaces:**

1. **What is interfaces?**

Ans:1.Interface is a keyword.

1. We can declare methods signatures only but not implementations.
2. By default all interface methods are abstract.

4.We can declare variables inside the interface all are public static final.

5 One class can implements more than one interface.

6.One interface can extend other interface.

1. **How can we declare interfaces?**

Ans:We can declare interface by interface keyword.

1. **What is the default methods of all interfaces?**

Ans: Abstract.

1. **Can we declare variables inside interface?**

Ans: We can declare variables inside the interface all are public, static ,final.

1. **Which keyword is used to inherit interface to class?**

Ans: Implements keyword is used to inherit interface to class.

1. **Which keyword is used to inherit from one interface to another?**

Ans: Using extends keyword ,One interface can extend other interface.

1. **Can class implements more than one interface?**

Ans: One class can implements more than one interface.

1. **Can we create the object for interface?**

Ans: We cannot create objects to interface, but we can create references to interface.

1. **What happens if we didn’t override all the interface methods?**

Ans: Compile time error.

1. **Does java support multiple inheritance?**

Ans: Yes, Java will support Multiple inheritance through interface.

1. **What are the marker interfaces?**

Ans: Marker interfaces are java.io.serializable,cloneable.

**Abstract:**

1. **what is abstract class?**

Ans: Abstract is a keyword.Abstract class having abstract methods and concrete(implemented) methods. If any class having one abstract method that class should be declare as abstract keyword, otherwise the class will be showing compile time error.

1. **What methods the abstract class consists of?**

Ans: Abstract class having abstract methods and concrete(implemented) methods.

1. **Can abstract class extends other class?**

Ans: Yes,abstract class can be extended in other classes but the abstract class methods should be override in that class.

1. **Can we create object to abstract class?**

Ans: We cannot create object to abstract class but we can create reference to abstract class.

1. **How can we access abstract class?**

Ans: We cannot create objects to abstract class but we can create reference to abstract class to access them.

1. **What happens if extends abstract class didn’t override all abstract methods?**

Ans: If any class extends abstract class, that class should be override all abstract all methods, otherwise the class will be showing compile time error.

1. **Can we declare abstract class without abstract methods?**

Ans: Yes***,***We can declare abstract class with zero abstract methods.

1. **Can we create constructor to abstract class?**

Ans: Yes.

1. **How can we access abstract class constructor?**

Ans: we can access abstract class constructor through sub class,object creation.

**HAS-A-RELATIONSHIP:**

1. **What is HAS-A relationship?**

Ans: IS-a relation is through object creation.

**IS-A-RELATIONSHIP:**

1. **What is IS-A relationship?**

Ans: HAS-A relation is through inheritance.