**Core Java Concept Questions**

**OOPS**

1. **What is OOPS ?**

OOPS is object oriented programming which is concept based language.

OOPS features are security,code reusability, enhancement and readability .

Main pillar in OOPS are

* Encapsulation
* Inheritance
* Polymorphism
* Abstraction

1. **What is Encapsulation ?**

It is a mechanism binding attributes and methods together inside a class and object creation

1. **What is Inheritance ?**

Inheritance means reusability of existing functionalities from superclass to subclass

Java won’t support multiple inheritance through classes(extends)

Java will support multiple inheritance through interfaces(implements)

Java will support multilevel inheritance through classes

1. **Will Java support Multiple Inheritance through classes ?**

Java won’t support multiple inheritance through classes(extends)

Java will support multiple inheritance through interfaces(implements)

1. **What is Polymorphism ?**

Polymorphism is doing a task in many ways. There are two types of polymorphism

1. Compile time polymorphism - Method Overloading
2. Runtime polymorphism - Method Overriding
3. **What is Overloading ?**

* In overloading methods are same
* Parameters are different
* Return type may or may not same
* Happens in same class or different class

1. **What is Overriding ?**

* In overriding methods are same
* Parameters are same
* Return type is also same
* Happens in different classes through inheritance

1. **What is Abstraction ?**

Abstraction means hide the complex implementation details of a class and expose only the essential features to the user.

**Constructor**

1. **What is a Constructor ?**

* Constructor is a method class name and method name should be the same .
* Constructors are mainly for initialization.
* There are two types of constructors
  + Default Constructor
  + Parameterized Constructor

1. **How can we create a Constructor ?**

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 ?**

We can access the Constructor while creating objects.

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

They are 2 types of Constructors that are : Default constructors and Parameterized constructors.

1. **Can we declare the constructor as void ?**

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

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

this is a keyword always pointing to the instance variable.

1. **What is the super keyword in Constructor ?**

super is a keyword always called the super class constructor.

1. **What is a Copy constructor ?**

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

**Static**

1. **What is Static ?**

Static is a Keyword. We can declare static as Methods & Variables.If Static variable is modified it will reflect globally.

1. **How can we access static variables and static methods ?**

We can access static variables and static methods directly through classname.variableName classname.methodName.

1. **What is a static block ?**

Whenever a class is loaded into JVM , first static blocks are initialized . Static blocks are used in databases and networking .

1. **Can static methods access non static methods ?**

No static methods can not access non static methods . They can only access static methods

1. **Can non-static methods access static methods ?**

Non static methods can static methods

**Final**

1. **What is final ?**

final is a Keyword. We can declare final as Variables, Methods & Class. Final variables can not be modified. Final methods can not be Overridden. Final Class can not be extended (Inherited).

1. **How can we declare a final?**

We can declare final as Variables, Methods & Class.

1. **Can final variables be modified?**

No final variables cannot be modified.

1. **Can final methods override?**

No final methods cannot be overridden.

1. **Can the final class inherit?**

No final methods cannot be inherited.

**Strings**

1. **What is a string ?**

In Java , a string is a sequence of characters , often used to represent text. Strings are objects in java and are instances of the string class, which is part of the java.lang package. Strings are immutable

1. **How many types of strings are there ?**

There are 3 types of Strings

* String class
* String Buffer
* String Builder

1. **What is String class ?**

* String class is immutable
* String is final class
* String having methods
* All string class methods are nonsynchronized

1. **What is String Buffer ?**

* StringBuffer is mutable
* StringBuffer is final class
* StringBuffer having methods
* All stringbuffer methods are synchronized

1. **What is Stringbuilder ?**

* StringBuilder is mutable
* StringBuilder is final class
* StringBuilder having methods
* All stringbuilder methods are nonsynchronized

1. **Explain the difference between equals method and == operator ?**

equals() - used to the check the content

== operator is used to the memory location

1. **What are string class methods? Explain them ?**

length() - gives the length of the string

toLowerCase() - converts the string to lowercase

toUpperCase() - converts the string to uppercase

trim() - remove the spaces

1. **What is meant by string concatenation ?**

Two strings are added using + operator or concat().

1. **What are string builder methods? Explain them ?**

* append() - adds a string to the end of the current string
* insert() - adds a string to a specific position in the current string
* replace() - replace a substring with a new string
* delete() - removes a substring from the current string
* reverse() - reverse the order of characters in the current string
* capacity() - returns the current capacity of the buffer
* length() - return the current length of the string

1. **What are string buffer methods? Explain them ?**

* append() - adds a string to the end of the current string
* insert() - adds a string to a specific position in the current string
* replace() - replace a substring with a new string
* delete() - removes a substring from the current string
* reverse() - reverse the order of characters in the current string

**Immutable**

1. **Can we make class immutable ?**

We can make class as immutable by declaring it as final class and attributes as private and final

1. **Can we create our own immutable class ?**

We can declare class as final

The class is declared as final so that it cannot be subclassed

**Interface**

1. **What is the interface ?**

* Interface is a keyword
* We can declare methods signatures only but not implementations.
* By default all interface methods are abstract.
* We can declare variables inside the interface all are public static final.
* One class can implement more than one interface.
* One interface can extend another interface.

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

We can declare interface by interface keyword.

1. **Can we declare variables inside the Interface?**

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

1. **Which Keyword is used to inherit Interface to class?**

Implements keyword is used to inherit interface to class

1. **Can we create an object to interface?**

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

1. **Does Java support Multiple Inheritance ?**

Yes, Java will support Multiple inheritance through interface.

**Abstract**

1. **What is an abstract class ?**

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

**2. What methods does the abstract class consist of?**

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

**3. Can an abstract class extend another class?**

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

**4. Can we create objects for abstract class?**

We cannot create objects to abstract class but we can create reference to abstract class.

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

We cannot create objects to abstract classes but we can create references to abstract classes to access them.

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

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

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

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

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

Yes.