Core Java Concept Questions

OOPS:

1. What is OOPS?
-> OOPS can be defined as Object Oriented Programming. Encapsulation, Inheritance, Polymorphism, Abstraction known as four pillars.
2. What is Encapsulation?
-> Encapsulation is binding data variables and methods together in a class.
3. What is Inheritance?
-> Reusability of existing functionalities from super class to sub class.
4. Is Java will support Multiple Inheritance through classes?
-> Yes. Java will support Multiple Inheritance through classes.
5. What is Polymorphism?
-> Single entity shows in different forms.
6. What is over loading?
-> Having same method names with different parameters, return type may or may not be same in same class.
7. What is overriding?
-> Having same method names with same method signature, same return type in different classes.
8. Difference between overloading and overriding?
-> Method signature will be different in overloading, method signature will be same in overriding.
9. What is Abstraction?
-> Hiding implementation details and showing only necessary functionalities.

Constructor:

1. What is Constructor?
-> Constructor is a special method mainly used for initialization of variables.
2. How can we create constructor?
-> We can create constructor by using same name as class.
3. How can we access constructor?
-> We can access constructor while creation of object.
4. How many ways we can create constructor?
-> We can create constructor by using parameters (parametarized constructor) and no parameters (default constructor).
5. Can we declare constructor as void?
-> No, we can’t declare constructor as void. If we declare it as void it will be considered as normal method.
6. What is this keyword in constructor?
-> this keyword refers to instance variables.
7. What is super keyword in constructor?
-> super keyword refers to super class constructor.
8. What is copy constructor?
-> copies an object using another object of same class.

Static:

1. What is static?
-> static is a keyword.
2. How can we declare static?
-> static can be declare as variable, method.
3. What is static block? What is the use of static block?
-> when a class loaded into JVM, static block is initialized first only.
4. Can static methods access non static methods?
-> No, static methods cannot access static methods.
5. Can non static methods access static methods?
-> Yes. Static methods can access non static methods.
6. How can we access static variables and methods?
-> static variables can be called by classname.variable and static methods can be accessed as classname.method.

Final:

1. What is final?
-> final is a keyword.
2. How can we declare final?
-> final can be declared as variable, method, class.
3. When can we initialize final variables?
-> final variables can be initialized while constructor calling.
4. Is final variables can be modified?
-> No. Final variables cannot be modified.
5. Can final methods be overrided?
-> No, final methods can't be overrided.
6. Can final class be inherited?
-> No, final class cannot be inherited.

Strings:

1. What is String class?
-> String is a final class and Immutable.
2. Why Strings are Immutable?
-> When we create strings, it will be constant. If we tried to modify the string, it will create another memory location, existing memory will be eligible for garbage collection.
3. How can we declare strings?
-> We can declare strings by two ways:
    -by using string literals
    -by using new keyword
4. What is String pool?
-> When we create a string using literals it's memory location will be created in string pool.
5. Difference between == and equals() method?
-> == operator compares references of two strings whereas equals() method compares content of the strings.

6. What is the nature of String methods?
-> String methods are Non synchronized.
7. What are String methods?
-> String have methods like length(), toLowerCase(), toUpperCase(), trim().
8. What is String Buffer?
-> String Buffer is final class and mutable.
9. Is String Buffer is mutable?
-> yes, String Buffer is mutable, we can modify string buffer.
10. What is mean by Thread safe?
-> Thread safe means methods will be executed step by step(synchronized).
11. Are String Buffer methods are synchronized or non synchronized?
-> String Buffer methods are synchronized.
12. What are String Buffer methods?
-> String Buffer methods are append(), insert(), delete(), replace(), length(), capacity().
13. What is String Builder?
-> String Builder is final class and mutable.
14. Is String Builder mutable?
-> yes, String Builder is mutable, we can modify string buffer.
15. What is the nature of String Builder methods?
->String Builder methods are non- synchronized.
16. When is String Builder introduced?
-> String Builder introduced after JDK 1.5
17. Strings are part of which package?
-> Strings are part of java.lang package.

Interfaces:

1. What is Interface?
-> Interface is a keyword. We can declare method signatures only not implementations.
2. Is Java support Multiple inheritance through Interfaces?
-> Yes, java support multiple inheritance through interfaces.
3. Can we create object to an Interface?
-> No, we cannot create object to an interface. But we can create reference to interface.
4. What type of methods will be there in Interface?
-> Interface contains only abstract methods.
5. What is default type of variables in Interface?
-> In interface, variables are public static final by default.
6. Can a class implements an interface?
-> Yes, a class can implements an interface.
7. Can an Interface extends another Interface?
-> Yes, a interface can extends another interface.
8. Is it necessary to override all the methods of interface?
-> Yes, if any class implements an interface, then all methods should be overrided in that class, otherwise it will shows compile time error.

Abstract class:

1. What is Abstract?
-> Abstract is a keyword.
2. How an abstract class is declared?
-> If any class contains atleast one abstract method, then that class should be declared by abstract keyword.
3. What type of methods will be there in abstract class?
-> Abstract class contains both abstract and concrete methods.
4. Is it necessary to override all the abstract methods which were there in abstract class?
-> yes, if any class extends an abstract class, then that class should override all abstract methods, otherwise it will shows compile time error.
5. Can we create object to abstract class?
-> No, we cannot create object to abstract class. But we can create reference to abstract class.
6. Can we declare a class as abstract, if it has zero abstract methods?
-> yes, we can declare a class as abstract even if it has zero abstract methods, but we cannot create object to that.
7. Can we create constructor to abstract class?
-> Yes, we can create constructor to abstract class.
8. How can we access abstract class constructor?
-> we can access abstract class constructor through sub class object.