1) What is constructor in Java?

A: a constructor in java is a block of code similar to a method that's called when an instance of an object is created.

2) What is the purpose of default constructor?

A: if there are no constructors, java compiler will add a default constructor and calls super class constructor, which is to initialize the internal state of an object so that the code creating an instance will have a fully initialized, usable object.

3) Does constructor return any value?

A: A constructor doesn't have a return type.

4)Is constructor inherited?

A: Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.

5) Can you make a constructor final?

A: static members belong to a class, but the constructor is needed to create an object. An abstract class is a partially implemented class, which contains abstract methods to be implemented in child class. final restricts modification: variables become constant, methods can't be overridden, and classes can't be inherited.

6) What is static variable?

A: a static variable is a variable that has been allocated statically so that its lifetime or "extent" extends across the entire run of the program.

7) What is static method?

A: a static method belongs to the class rather than object of a class. A static method invoked without the need for creating an instance of a class. static method can access static data member and can change the value of it.

8) Why main method is static?

A: This is neccesary because main() is called by the JVM before any objects are made. Since it is static it can be directly invoked via the class. Similarly, we use static sometime for user defined methods so that we need not to make objects. void indicates that the main() method being declared does not return a value.

9) What is static block?

A: Static block is mostly used for changing the default values of static variables.This block gets executed when the class is loaded in the memory. A class can have multiple Static blocks, which will execute in the same sequence in which they have been written into the program.

10)What is difference between static (class) method and instance method?

A: To be said frankly the main difference is an instance method belongs to an object while a static method belongs to the class.

Which means to access an instance method you need to make an object out of a java class and the object has instance methods and instance variables which are accessible through the object. (object reference)

The static methods and variables do not need an object. They are accessed through the class. (class reference)

11) What is this in java?

A: this is a keyword in Java. It can be used inside the Method or constructor of Class. It(this) works as a reference to the current Object whose Method or constructor is being invoked. The this keyword can be used to refer to any member of the current object from within an instance Method or a constructor.

12)What is Inheritance?

A: Inheritance refers to a feature of Java programming that lets you create classes that are derived from other classes. A class that's based on another class inherits the other class. The class that is inherited is the parent class, the base class, or the superclass.

13) Which class is the superclass for every class.

A: Object class.

14) Is multiple inheritance is supported in java?why?

A: No, It is just to remove ambiguity, because multiple inheritance can cause ambiguity in few scenarios.

15) What is super in java?

A: super is a keyword. It is used inside a sub-class method definition to call a method defined in the super class. Private methods of the super-class cannot be called. Only public and protected methods can be called by the super keyword.It is also used by class constructors to invoke constructors of its parent class.

16)What is method overloading?

A: Method Overloading is a feature that allows a class to have two or more methods having same name, if their argument lists are different. In the last tutorial we discussed constructor overloading that allows a class to have more than one constructors having different argument lists.

17) Can we overload main() method?

A: yes we can overload main method. main method must not be static main method. Yes, main method can be overloaded. Overloaded main method has to be called from inside the "public static void main(String args[])" as this is the entry point when the class is launched by the JVM.

18) What is method overriding?

A: The ability of a subclass to override a method allows a class to inherit from a superclass whose behavior is "close enough" and then to modify behavior as needed. The overriding method has the same name, number and type of parameters, and return type as the method that it overrides.

19) Can we override static method?Why not?

A: Static methods cannot be overridden because they are not dispatched on the object instance at runtime. The compiler decides which method gets called. ... No,Static methods can't be overriden as it is part of a class rather than an object. But one can overload static method.

20) Difference between method Overloading and Overriding.

A: Overloading occurs when two or more methods in one class have the same method name but different parameters. 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.

21) What is final variable/final method/final class?

A: In Java we use final keyword to variables to make its values not to be changed.

A final method cannot be overridden or hidden by subclasses. This is used to prevent unexpected behavior from a subclass altering a method that may be crucial to the function or consistency of the class.

You can declare some or all of a class's methods final. You use the final keyword in a method declaration to indicate that the method cannot be overridden by subclasses.

22) Can you declare the main method as final?

A: Sure, it can be declared final! It doesn't matter if it is declared final, the JVM will still find it and run it, and the compiler doesn't care. Yes We can declare main method final. If we make this final it can not be override.Hence we can't use it in its child class.

23) What is Runtime Polymorphism?

A: The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object. Dynamic Binding/Runtime Polymorphism : Run time Polymorphism also known as method overriding. In this Mechanism by which a call to an overridden function is resolved at a Run-Time.

24) What is abstraction?

A: Abstraction is the process of abstraction in Java is used to hide certain details and only show the essential features of the object. In other words, it deals with the outside view of an object (interface).

25) What is the difference between abstraction and encapsulation?

A: Encapsulation is wrapping, just hiding properties and methods. Encapsulation is used for hide the code and data in a single unit to protect the data from the outside the world. Class is the best example of encapsulation. Abstraction refers to showing only the necessary details to the intended user.

26) What is abstract class?

A: Abstract classes are classes that contain one or more abstract methods. An abstract method is a method that is declared, but contains no implementation. Abstract classes may not be instantiated, and require subclasses to provide implementations for the abstract methods.

27) Can there be any abstract method without abstract class?

A: Yes we can have an abstract class without Abstract Methods as both are independent concepts. Declaring a class abstract means that it can not be instantiated on its own and can only be sub classed. Declaring a method abstract means that Method will be defined in the subclass.

28) Can you use abstract and final both with a method?

A: No, Abstract method cannot be declared as final. This is because, abstract method has to be overriden to provide implementation. If it is declared as final, it cannot be overridden.

29) Is it possible to instantiate the abstract class?

A: Abstract classes cannot be instantiated, means we can't create an object to Abstract class.

30) What is interface?

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

31) Can you declare an interface method static?

A: You can't define static methods in an interface because static methods belongs to a class not to an instance of class, and interfaces are not Classes.

32) Can an Interface be final?

A: An interface is a pure abstract class. Hence, all methods in an interface are abtract , and must be implemented in the child classes. So, by extension, none of them can be declared as final . In the context of Java 8 and default methods, this question has a new meaning.

33) What is difference between abstract class and interface?

A: 1.Main difference is methods of a Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. 2.Variables declared in a Java interface is by default final. An abstract class may contain non-final variables.

34) Can we define private and protected modifiers for variables in interfaces?

A: only public, static & final are permitted. You should use an abstract class instead of an interface if you want to declare its fields.

35) What is package?

A: A package is a namespace that organizes a set of related classes and interfaces. Conceptually you can think of packages as being similar to different folders on your computer. You might keep HTML pages in one folder, images in another, and scripts or applications in yet another.

36) What is Exception Handling?

A: Exception Handling is a mechanism to handle runtime errors such as ClassNotFound, IO, SQL, Remote etc.

37) What is difference between Checked Exception and Unchecked Exception?

A: In Java exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under throwable is checked. Consider the following Java program. It compiles fine, but it throws ArithmeticException when run. The compiler allows it to compile, because ArithmeticException is an unchecked exception.

38) What is difference between throw and throws?

A: In java, Exception is handled by throwing object of appropriate Exception class. ... In Java , throw keyword is used to explicitly throw an Exception.Wherever you want that exception occurs you can use throw. whereas, throws is used with method signature.

39) What is finally block?

A: Java finally block is a block that is used to execute important code such as closing connection, stream etc. Java finally block is always executed whether exception is handled or not. Java finally block follows try or catch block.

40) What is the meaning of immutable in terms of String?

A: String is immutable means that you cannot change the object itself, but you can change the reference to the object. When you called a = "ty" , you are actually changing the reference of a to a new object created by the String literal "ty" .

41) Why string objects are immutable in java?

A: Java Developers decide Strings are immutable due to the following aspect design, efficiency, and security.

Design Strings are created in a special memory area in java heap known as "String Intern pool". While you creating new String(Not in the case of using String() constructor or any other String functions which internally uses String() constructor for creating new String object; String() constructor always create new string constant in the pool unless we call the method intern()) variable it searches the pool to check whether is it already exist. If it is exist, then return reference of the existing String object. If the String is not immutable, changing the String with one reference will lead to the wrong value for the other references.

Security String is widely used as parameter for many java classes, e.g. network connection, opening files, etc. Were String not immutable, a connection or file would be changed and lead to serious security threat. Mutable strings could cause security problem in Reflection too, as the parameters are strings.

Efficiency The hashcode of string is frequently used in Java. For example, in a HashMap. Being immutable guarantees that hashcode will always the same, so that it can be cached without worrying the changes.That means, there is no need to calculate hashcode every time it is used.

42) How many ways we can create the string object?

A: You can use the StringBuilder Class

Or you can use String class

You can use String.format method

Or you can use .toString method to create strings.

43) What is the difference between String, StringBuffer and StringBuilder ?

A: StringBuffer and StringBuilder have the same methods with one difference and that's of synchronization. StringBuffer is synchronized( which means it is thread safe and hence you can use it when you implement threads for your methods) whereas StringBuilder is not synchronized( which implies it isn't thread safe).

44) How can we create immutable class in java ?

A: Declare the class as final so it can’t be extended.

Make all fields private so that direct access is not allowed.

Don’t provide setter methods for variables

Make all mutable fields final so that it’s value can be assigned only once.

Initialize all the fields via a constructor performing deep copy.

Perform cloning of objects in the getter methods to return a copy rather than returning the actual object reference.

45) What is the purpose of toString() method in java ?

A: The toString() method returns the string representation of the object. If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object etc. depends on your implementation.

46) How can we create immutable class in java ?

the same as question No.44

47) What is the purpose of toString() method in java ?

the same as question No.45

48) What is Garbage Collection?

A: The garbage collection is a program which runs on the Java Virtual Machine which gets rid of objects which are not being used by a Java application anymore. It is a form of automatic memory management.

49) What is the purpose of finalize() method?

A: The java.lang.Object.finalize() is called by the garbage collector on an object when garbage collection determines that there are no more references to the object. A subclass overrides the finalize method to dispose of system resources or to perform other cleanup.

50) What is difference between final, finally and finalize?

A: Final class can't be inherited, final method can't be overridden and final variable value can't be changed. Finally is used to place important code, it will be executed whether exception is handled or not. Finalize is used to perform clean up processing just before object is garbage collected.

51) Can you access the private method from outside the class?

A: use setAccessible(true) on your Method object before using its invoke method. First you gotta get the class, which is pretty straight forward, then get the method by name using getDeclaredMethod then you need to set the method as accessible by setAccessible method on the Method object.

52) What is singleton class?

A: Here, ClassicSingleton class employs a technique known as lazy instantiation to create the singleton; as a result, the singleton instance is not created until the getInstance() method is called for the first time. This technique ensures that singleton instances are created only when needed.

53) What do you mean by Access Modifier? Types?

A: Java provides a number of access modifiers to set access levels for classes, variables, methods, and constructors. The four access levels are −

Visible to the package, the default. No modifiers are needed.

Visible to the class only (private).

Visible to the world (public).

Visible to the package and all subclasses (protected).

54) Explain Singleton Pattern? Basic conditions required to achieve it?

A: Singleton pattern restricts the instantiation of a class and ensures that only one instance of the class exists in the java virtual machine. The singleton class must provide a global access point to get the instance of the class. Singleton pattern is used for logging, drivers objects, caching and thread pool.

The Singleton's purpose is to control object creation, limiting the number of objects to only one. Since there is only one Singleton instance, any instance fields of a Singleton will occur only once per class, just like static fields.

55) What is an enumeration?

A: enums are implicitly final subclasses of java.lang.Enum. if an enum is a member of a class, it's implicitly static. new can never be used with an enum, even within the enum type itself. name and valueOf simply use the text of the enum constants, while toString may be overridden to provide any content, if desired.

56) What is Downcasting?

A: downcasting means casting to a subtype, downcasting must be manually done by the programmer.

57) What is type casting in Java?

A: Type casting in Java is to cast one type, a class or interface, into another type i.e. another class or interface. Since Java is an Object oriented programming language and supports both Inheritance and Polymorphism, It’s easy that Super class reference variable is pointing to SubClass object but the catch here is that there is no way for Java compiler to know that a Superclass variable is pointing to SubClass object. Which means you can not call a method which is declared in the subclass. In order to do that, you first need to cast the Object back into its original type. This is called type casting in Java.

58) Give examples of Data encapsulation, Polymorphism and inheritance used in your project?