# CoreJava Prep

1. What are anonymous classes ?

Ans - Anonymous classes are inner classes with no name. we have to declare and instantiate anonymous classes in a single expression at the point of use. While defining anonymous class, we may either extend an existing class or implement an interface.

3. How do you start new Thread in Java ?

There are 2 ways by which new thread can be created. One way is that you call start() method of Thread class. It is very important to remember that if we call run() method of Thread class then new thread will not be created. So always call Thread.start() method to create a new thread.

4. Thread.currentThread().getId() - What is this method call ?

Ans - Thread.currentThread() method returns reference of current thread and its getId() method returns thread id.

5. What do you understand by annotation @FunctionalInterface ?

Ans - A Functional interface is an interface which contains only one abstract method. However it can contain any number of default methods. From Java 8 onwards lambda expression can be used to represent instance of a functional interface.

6. What is a protected data member in Java ?  
Ans - Protected data member and method are only accessible by the classes of the same package and the subclasses present in any package.

7. What are differences between Java interface and abstract classes ?  
Ans -

* Java interfaces are implicitly abstract and cannot have implementation. Java abstract class can have instance methods that has default implementation.
* Members of Java interface are by default public, while members of abstract class can use keywords like public / private / protected
* An interface can extend another Java interface while an abstract class can extend another Java class and can implement multiple interfaces.

8. What is compareTo() method in Java with example ?  
Ans - compareTo() method is used to perform natural sorting on string. Natural sorting means the sort order which applies on the object, e.g., lexical order for String, numeric order for Sorting integers, etc.

Lexical order is nothing but alphabetically order. compareTo methods does a sequential comparison of letters in the string that have the same position.

compareTo is defined in interface java.lang.Comparable.

Strings in Java

* In Java, string is basically an object that represents sequence of char values. An array of characters works same as Java string
* Java String class provides a lot of methods to perform operations on string such as compare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc
* The java.lang.String class implements Serializable, Comparable and CharSequence interfaces.
* The CharSequence interface is used to represent the sequence of characters. String, StringBuffer and StringBuilder classes implement it. It means, we can create strings in java by using these three classes.

Q. How to create String object ?  
Ans - There are two ways to create String object,

* By String literal
* By new keyword

Q. What can we learn while learning String ?

* Concept of String
* Immutable String
* String Comparison
* String Concatenation
* Concept of Substring
* String class methods and its usage
* StringBuffer class
* StringBuilder class
* Creating Immutable class
* toString() method
* StringTokenizer class

Q. What is a final keyword in Java ?  
Ans - final keyword is used in different contexts. First of all, final is a non-access modifier applicable only to a variable, a method or a class.Following are different contexts where final is used.

* Final variable - To create constant variable
* Final methods - To prevent method over-riding
* Final classes - To prevent inheritance

Important note about "final" variables in Java :-

When a variable is declared with final keyword, its value can’t be modified, essentially, a constant. This also means that you must initialize a final variable. If the final variable is a reference, this means that the variable cannot be re-bound to reference another object, but internal state of the object pointed by that reference variable can be changed

Q. Note difference between "Initializer block" and "Static Initializer Block" in Java.  
Ans - Initializer block is used to initialize instance variable, while "Static initializer block" is used to initialize static variables.

1. Explain substring.

public String substring(int startIndex)

startIndex (inclusive)

public String substring(int startIndex, int endIndex)  
startIndex: inclusive

endIndex: exclusive

Important Note :-String class implements three interfaces,

* Serializable
* Comparable
* CharSequence

CharSequence interface is used to represent sequence of characters. String, StringBuffer and StringBuilder class implements this interface.

* String class of Java is immutable.
* For Mutable String we need to use StringBuffer and StringBuilder class

Q. What is a functional interface in Java ?  
Ans - An interface which contains only one i.e single abstract method is known as Functional Interface.  
Lambda is mainly used to implement functional interfaces. Prior to Java8 bellow are two famous functional interfaces are,

* Comparator interface
* Runnable interface

Q. What is an Lambda expression ?

Ans - Lambda is equivalent to function (method) without a name.  
Lambda is also referred as anonymous function. A function will have these basic properties

* Method parameters
* Method body
* Return type

1. Lambda is not tied with any class like any regular method.
2. Lambda can be assigned to a variable and passed around. Below is the lambda expression, () -> {}

() - Parenthesis is a symbol for method parameters / Also called Lambda Input parameters  
 {} - Curly braces represents method body / Also called Lambda body

Q. In context of functional interfaces what is full form of SAM ?  
Ans - Full form of SAM is "Single Abstract Methods Interface".

Q. Can we have more than one class in a Java file ?  
Ans :- You can have multiple classes within a class. They are called Inner Classes or nested classes. You can even have multiple class definitions in a single .java file without one being nested in another (provided that only one is public, because a public class has to be declared in a file named after it.