**Class in Java**

**Q 1: What is Class?**

**Ans:** Class is a blueprint of its object.  It has state and behavior, (by state we mean variable and by behavior we mean method).It describes a state and behavior of that object.

**Q: What is the base class of all classes?**

It is a Object Class.

Few methods of Object classes are:   finalize(), notify(), notifyAll(), equals(), wait(), wait(x), wait(x, y), getClass(), hashcode()

**Q: Can a source file contain more than one class declaration ?**

**Ans:** Yes, a single source file can contain any number of Class declarations but only one of the classes can be declared as public.

**Q: Can a class declared as private or protected? What is the impact of declaring a method as final?**

* Not possible.
* A method declared as final can't be overridden. A sub-class can't have    the same method signature with a different implementation.

**Q: I don't want my class to be inherited by any other class. What should i do?**

* You have to  declared your class as final because final class cannot  be inherited.

**Q: Can you give few examples of final classes defined in Java API?**

* java.lang.String,  java.lang.Math are final classes.

**Q:Can a class be declared as static?**

We can not declare top level class as static, but only inner class can    be declared static.

**Example:** public class Test {

static class InnerClass {//is called inner class}}

**Q: Is declaring a main() method compulsory in all Java Classes ?**

It is not required for all classes but we need a class with main() method at least once to execute our project.

**Q:What is the difference between a static and a non-static inner class?**  
  Ans:A non-static inner class may have object instances that are associated     with instances of the class's outer class. A static inner class does not have any object instances.

**Q: What is the difference between inner class and nested class?**

**Ans**:Nested classes are divided into two categories:  static and non-static.   Nested classes that are  declared static are simply called static nested  classes. Non-static nested classes are called inner classes.

**Q: What is the difference between a public and a non-public class?**  
**Ans:**A public class may be accessed outside of its   package. A non-public class may not be accessed  outside of its package.

**Q. What is final class? Ans -**  Ans:Final class can not be inherited by sub class.Only u can instantiate.

**Q: What is a object? :If you create a object it showing an error what could be the reason?**

Object is an instance of a class. Mainly it could be Constructor parameter did not match.

**Variable in Java**

**Q1)What is variable? What is Local ,Global and Static Variable?**

Variable is name of memory location.      Example:   int a(variable name)=5;

**Variable is of Three:**

     a) Local variable--- are those that are declared in methods, constructors, or blocks. Access modifiers cannot be used for local variables

     b) Global variable:---- Global variables - are declared outside the method and inside the class. They are also known as Instance variable.

**Global variable are two types**

1) Class variable(static keyword) and 2) Instance variable(No static Variavle)

**1)Static Variable- or class variables** are those which are created when the class is created.

Like public static int id;

Class variables also known as static variables are declared with the static keyword in a class, but outside a method, constructor or a block.

**2)Instance Variables** are those which are created when the object of a class is created.

Like public int id;

Instance variables are declared in a class, but outside a method, constructor or any block.

**Q2: What is the purpose of declaring a variable as final?**

A final variable's value can't be changed. final variables should be  initialized before using them.

**Q3: How is final different from finally and finalize()?**

Final is a modifier which can be applied to a class or a method or a    variable.final class can't be inherited, final method can't be overridden and   final variable can't be changed.  finally is an exception handling code section which gets executed whether an   exception is raised or not by the try block code segment.   finalize() is a method of Object class which will be executed by the JVM   just before garbage collecting object to give a final chance for resource   releasing activity.

**Q4:** **What is the difference** **between a Instance and a Class variable?**

 A.  When you declare a variable with a Static keyword in a Class level then that is called a Class variable.

      B.  When you declare a variable without static keyword in a Class level then it is called an Instance variable.

**Q5: What is a Static Variable ? Tell me the benefit of a Static Variable?**

When you use a Static keyword before any Variable then that is called a Static Variable. If you declare a Variable as a Static then that will work with change of value and you can also access that with the class name.

If you declare a Method as a Static then you can overload that Method but cannot override. You can also call that with the class name and only take static property.

**NOTE:** You cannot declare Static in a Class level. Static variable makes your program memory efficient (i.e it saves memory).

**Q6: How to define a constant variable in Java ?** The variable should be declared as static and final. So only one copy of the variable exists for all instances of the class and the value can't be changed also.

**Example:**   static final int MAX\_LENGTH = 50;

**Q7: What is the importance of static variable?**

static variables are class level variables where all objects of the   class refer to the same variable. If one object changes the value then the change gets reflected in all the objects and can be called by class name.

 **Q: Can we declare a static variable inside a method?** Static variables are class level variables and they can't be declared  inside a method. If declared, the class will not compile.

**Q: If a variable is declared as private, where may the variable be accessed?**  
  
A private variable may only be accessed within the class in which it is declared.If you want to give access to another class use getter and setter  method.getter mean read and setter mean write.

**Q: To what value is a variable of the boolean type automatically initialized?**  
  
  Ans:The default value of the boolean type is false.

**Q.What happens to a static variable that is defined within a method of a class ?**  
  
  Ans:You can not declared static variable in a method.

**Q:To what value is a variable of the String type automatically initialized?**  
  
  Ans:The default value of an String type is null.  
  
**Q. What is a transient variable?**  
  
  Ans:Transient variable is a variable that may not be serialized.  
  
**Q: What is the default value of an object reference declared as an instance variable?**  
  
  Ans:The default value will be null unless we define it  explicitly.

**Q: What happens if you don't initialize an instance variable of any of the primitive types in Java?**  
  
Ans:Java by default initializes it to the default value for that primitive type. Thus an int will be initialized to 0(zero), a boolean will be initialized to false.

**Q: How to get a Private Variable in another Class ?**  By using getter setter method

   (getter mean read, setter mean write) **Example:**

                          private int a=8;

                           int getA()

                         {

                                return a;

                        }

                       int setA(int x)

                      {

                              a=x;

                             return a; }

**Method in Java**

**Q: What is a method?    Ans:** It is a collection of statements which are grouped together to perform an operation.

**Q: Is declaring a main() method compulsory in all Java Classes ?**

 It is not required for all classes but we need a class with main() method at least once to execute our project.

**Q: What is the return type of a main() method ?** A main() method doesn't return anything, hence it is declared void.

**Q: When will you define a method as static and what is restriction?**

When a method needs to be accessed even before the creation of the   object of the class then we should declare the method as static. A static method should not refer to instance variables without creating an   instance and cannot use "this" operator to refer the instance.

**Q: What is the impact of declaring a method as final?**

A method declared as final can't be overridden. A sub-class can't have    the same method signature with a different implementation.

**Q: Why is the main() method declared as a static ?**

**Ans:**      main() is called by the JVM even before the instantiation of the class, hence it is declared as static.

**Q: What is the argument(args) of a main() method ?      Ans:**     A main() method accepts an array of String objects as a argument. Java main method is the entry point of any Java program. you can only change the name of String array Argument, for example , you can change args to my String Args.

**Q: Does the order of declaring public and static have any effect in main() method ?**

  No, it doesn't matter but void should always come before main().

**Q: What restrictions are placed on method overloading?**    
  Ans: Method name must be same, parameter must be different, access modifier and return type may be different, and it happens in the same class.  
  
**What does it mean that a method or field is "static"?**  
  
  Ans: Static variable means is called class variable.Static variable always work with change value.Static variable u can call with class name means without creating an object. Static methods can be   referenced with the name of the class  rather  than the name of a particular   object of the class (though that works too). Static method can be overload but u can not override.

**.Q: What are the restrictions placed on method overriding?**  
  
   Ans: Overridden methods must have the same name,  argument list, and return type. The overriding  method may not limit the access of the method it  overrides. The overriding method may not throw any exceptions that may not be thrown by the overridden  method.  
  
**Q:What is difference between method overloading and overriding?**  
  
Ans: overloading: Overloading happen in a same class. Same method name with different parameter (size/type). Access modifier and return type may be changed.Compile time polymorphism.  
  
overriding: Overriding happens between parent class and child class. The body will be changed but the access modifier, method name, return type and parameter cannot be changed. Run time polymorphism.

### Interfaces

**Q1)What is an interface? Why we need interface?** An interface is a collection of abstract methods.

**Features of interface-** 1) Interface cannot be instantiated 2) An interface does not contain any constructors.3) All the methods in an interface are abstract.4) Interface class contain only unimplemented method. 5) In interface class, interface keyword should be used.

**Q2. Why we need to use ? 1)** **It is** useful when you cannot **use** class inheritance, because interface class can implement **multiple** interfaces.so we get multiple inheritance advantage. 2) Interfaces are useful in a situation where all properties need to be implemented. Interface class contain only unimplemented method.

### Q3- Differentiate an Interface and an Abstract class. Also discuss their similarities.

An abstract class contain abstract method(without body) and Non abstract method. It doesn’t supports multiple inheritance. it can have static, non static, final and Nonfinal variable. Abstract classes can have a partial implementation. **Example-abstract class Animal**{ abstract void eat()}.In abstraction,when unimplemented methods implemented in child class, **overriding concept is** happened.

**On the other hand**, Interface contain only abstract methods. It supports multiple inheritance . It can only have static and final variable. It fully abstract. Example- interface Animal{ void eat();}

In case of inheritance, we use implements keyword.

---**Similarities** :  
\* Neither abstract classes nor interfaces can be instantiated.

**Q4: What modifiers are allowed for methods in an Interface and abstract ?-**  Ans: Only public and abstract modifiers are allowed for methods in  interfaces. . Abstract can have static, non static, final and Nonfinal variable.

**Q5: Lets say you have a class which implement a interface but class did not implements all methods then is that possible you can create an object of that class?**

No because regular class can not take unimplemented methods so that you   have to declared that class as abstract. As we know abstract class can not be   instantiated. **Q: Can a method inside a Interface be declared as final? Ans--**No not possible. Public and abstract are the only applicable modifiers     for method declaration in an interface.

**Q6: Why is an Interface be able to extend more than one Interface but a Class can't extend more than one Class?** Basically Java doesn't allow multiple inheritance, so a Class is  restricted to extend only one Class. But an Interface is a pure abstraction   model and doesn't have inheritance hierarchy like classes(do remember that   the base class of all classes is Object). So an Interface is allowed to   extend more than one Interface.

### Q7.How can we define an interface?

In Java an interface just defines the methods and not implement them. Interface can include constants.  
A class that implements the interfaces is bound to implement all the methods defined in an interface.  
Example of Interface :

public interface sampleInterface }  
public void functionOne();  
public long CONSTANT\_ONE = 1000; }

### Q8.How do we use comparator and comparable interfaces?

• java.util.Comparator  
• java.util.Comparator:compares some other classes instances,  
• java.lang.Comparable  
• java.lang.Comparable:compares another object with itself.

**Q9: Can an Interface be declared as a final?** Ans: No.

**Q10: Can a class be defined inside an Interface?**   Ans: Yes it's possible.  
 **Q11: Can an Interface be defined inside a class?**  Ans Yes it's possible.  
**Q12: Can an Interface implement another Interface? Ans -**No not possible.

**Q13: Can an Interface extend another Interface? Ans-**Yes an Interface can inherit another Interface, for that matter an  Interface can extend more than one Interface.

**Q14.Can an inner class be built in an interface ?**

Yes, an inner class can be built in an interface.  
Example :

public interface xyz{  
static int p=0;  
void m();  
class c   
{  
c()  
{  
int q;  
System.out.println("inside");  
};  
public static void main(String c[])  
{  
System.out.println("inside ");}}}

### Q15.How interface is similar to a class? 1) Interface can have any number of methods, 2) It has same file extension as class(.java).

### How interface is different from a class? 1) you can’t instantiate an interface.(we can create object of class but can not create object of interface) 2) Interface doesn’t contain constructor(We can’t declare constructor of interface) 3) All the methods in interface are abstract. 4) An interface can’t have instance variables(We can’t declare instance variables),5) An interface can extend multiple interface. But class doesn’t support multiple inheritance.

**Q 16.In which kind of situation would an interface be extended by another interface ?**

Remember that any class that implements an interface must implement the method headings that are declared in that interface.  
If that particular interface extends from other interfaces, then the implementing class must also implement the methods in the interfaces that are being extended or derived from.  
As shown in the example above, if we have a class that implements the FourLegs interface,   
then that class must define the method headings in both the 'FourLegs' interface and the 'Body' interface.

### Q17.What is a marker interface?

Marker interface is an interface with no fields or methods in Java.   
Uses of marker interface are as follows:   
• We use marker interface to tell java compiler to add special behavior to the class implementing it.   
• Java marker interface has no members in it.   
• It is implemented by classes in order to get some functionality.   
For instance when we want to save the state of an object then we can implement serializable interface.

### Constructor

1. **What are constructors in Java?** --Constructor is a block of codes similar to the method. It must have the same name as that of the class. It has no return type and it is automatically called when an object is created. So we don’t need an object to a constructor as we need the object to call a function. **Syntax of Constructor.** Class Name( ){ }.

**There are two types of constructors:** Default constructor-(non-org cons)[that have no paramiter]

Parameterized constructor (A Con.that have parameter is known)

**2) Feature**s-- Constructor are very much similar to the method . But the basic differences is that 1) Constructor can’t return any value that’s why they don’t have any return type. 2) Constructor name must be same to the class name.3) Constructor will be called automatically when we create a new object. 5) Every time an object is created using the new() keyword, at least one constructor is called.

**3)When u need constructor? Can you overload or override a constructor?**

**We need constructor** 1) To construct or to  change a value of a variable then you can use constructor. i.e it eliminates placing the default values And 2) to initialize the object of a class. 3) To initialize the variables correctly.

### **Yes, Constructor is a block of code which execute at the time of object creation. When we want to change a value of a variable then we can use constructor and you have to change parameter type or parameter size. So to change parameter size or type we can overload a constructor. But you can’t override a constructor in java. Because constructor looks like a method but name should be as class name and no return value.**

**4)Distinction between Constructor and method.**

Java Constructor -**A constructor is used to initialize the state of an object. A constructor must not have a return type.** The constructor is invoked implicitly.. The constructor name must be same as the class name. **It is invoked at the time of object creation.Constructors cannot be abstract, final, static and synchronised while methods can be.**

**Java Method** A method is used to expose the behavior of an object. A method must have a return type. The method is invoked explicitly.The method name should not be same as the class name.

**Method invoking is optional.** Method can be abstract, final, static and synchronized.

**5) Does a class inherit the constructors of its super class?**   Ans: A class does not inherit constructors from any of its super classes.

**6)  When does the compiler supply a default constructor for a class?**  The compiler supplies a default constructor for a  class if no other  constructors are provided.

**7) How are this() and super() used with constructors?**  
-- this() is used to invoke a constructor of the same    class. super() is used to invoke a superclass  constructor.

8) Does abstruct class have Constructor or how to execute constructor of abstruct class ?

When we create object of class which is extending abstract class then it will call abstract class constructor through the sub class constructor. this is how the constructor of Abstract class will be executed.

**9) Can we define parameterized constructor in abstract class?** Yes, We can define parameterized constructor in sub class. But we need to make sure that class which is extending abstract class have a contractor and it should call super class parameterized contractor. So we can call super class parameterized constructor in subclass by using super()call.

**10) How are ‘this () ’ and super () used with Constructor?** --- this() is used to invoke a constructor of the same class( Current class & Global variable). And super() is used to invoke a super class constructor and parent class variable.

### ****Constructor overload and override****

### **We can overload the constructor**. Java allows having more than one constructor inside one class with the same name but different parameter. Constructor overloading is very much similar to the method overloading. But the basic differences is that 1) Constructor can’t return any value that’s why they don’t have any return type. 2) the Name of constructor must be same to the class name.3) Constructor will be called automatically when we create a new object. But we can not override a constructor in java.

### 4) It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

### The Java compiler provides a default constructor if you don't have any constructor in a class

1. **How to execute constructor of Abstract class?**  
    Ans: We need to extend the abstract class, where we have to create a constructor-through this constructor by the help of keyword ‘super’ we can reach parent class. Then we have to create object in another class then the constructor of extended class will be executed;

**Java Array**

1. **What is array and why we need array ?** Array is a container which holds same types of value and fixed size. it is when we need more than one value in a same variable.
2. **Fearures—1)**Array is a set of similar data type.2) Arrays objects store multiple variables with the same type.3)It can hold primitive types and object references.4) Arrays are always fixed in size 5)
3. **Are Arrays Primitive Data Types ? Ans:**     No, because in Java, Arrays are Objects.
4. **How to create an Array?** An Array is declared similar to how a variable is declared, but you need to add [] after the type. **Example:** **int** [] intArray;

We can declare Java array as a field, static field, a local variable, or parameter, like any other variable.

1. **Advantages and disadvantages of Array?**

**Advantages:**

* 1) We can represent huge number of values by using a single variable that’s why reusability of code will be improved. 2) Arrays can sort multiple elements at a time. 3) We can access an element of Array by using an index.

**Disadvantages:**

* Arrays are fixed in size ---1) To use array’s concept compulsory we should know the size in advance which may not possible always. 2) Once we create an array we can’t increase or decrease the size based on our requirement.it takes more memory space.
* Homogeneous data- Array can hold only homogeneous data elements.
* Underlying data structure-There is no underlying data structure that’s why readymade method support we can’t expect.

1. **Can we change the size of an array at run time?**

No we cannot change the array size. Though there are similar data types available which allow a change in size.

1. **Can you declare an array without assigning the size of an array?**  
   No we cannot declare an array without assigning size.

If we declare an array without size, it will throw compile time error

Example: marks = new int []; //COMPILER ERROR

1. **What is the default value of Array?**

Any new Array is always initialized with a default value as follows

* For byte, short, int, long – default value is zero (0).
* For float, double – default value is 0.0.
* For Boolean – default value is false.
* For object – default value is null.

1. **How to print element of Array?** Here is code to print element of the array.

|  |  |
| --- | --- |
| 1  2  3 | int schoolmarks [] = {25, 30, 50, 10, 5 };  System.out.println (Arrays.toString (schoolmarks)); // Output is [25, 30, 50, 10, 5] |

1. **When will we get ArrayStoreException?**

It is a runtime exception. For example, we can store only string elements in a String Array. If anybody tries to insert integer element in this String Array, then we will get ArrayStoreException at run time.

1. **Can we add or delete an element after assigning an array?**  Ans -No it is not possible.
2. **How to copy an array into another array?** Below four tricks are available in java to copy an array.
3. Using “For loop” b) Using “Arrays.copyOf()” method c) Using “System.arraycopy()” method
4. Using “clone()” method
5. **Can you tell me the class name of an array in Java?**

Array is an object. To retirve class name we can use getClass().getName() method on the object.

1. **Where is the memory allocated for arrays in Java?**

In Java, memory for arrays is always allocated on the heap as arrays in Java are objects.

1. **What is the two-dimensional array?** An array of an array in Java. We can declare them like  
   int[][] p = new int[3][3] which is a matrix of 3×3.
2. **Do we have 3-dimensional arrays in Java?** Yes, Java supports N-dimensional array.

Multi-dimensional array in Java is nothing but an array of array,

Example: 2-dimensional array is an array of 1-dimensional array.

1. **Can we make array volatile in Java? Ans-**Yes, we can make an array volatile in Java, but we only make the variable which is pointing to array volatile.
2. **What are “jagged” arrays in java?** Jagged Arrays are Arrays are containing arrays of different length. Jagged arrays are also known as multidimensional arrays.
3. **When ArrayIndexOutOfBoundsException occurs?** It is a run time exception. It will occur when the program tries to access invalid index of an array. Index higher than the size of the array or negative index.
4. **Is there any difference between int[] a and int a[]? Ans-**No difference both are the legal statement.
5. **int a[] = new int[3]{1, 2, 3} – is it a right way to declare arrays in java?** No. We should not mention the size of the array when we are providing the array elements.
6. **We know that Arrays are objects so why cannot we write strArray.length()?**

Arrays are object references like classes in Java. We can use the methods of Object like toString () and another one hashCode () against Array. Length is a data item of an array and so it is not a method. That’s why we are using strArray.length.

1. **How do we search a specific element in an array?**

We can use Arrays.binarySearch() method. This method uses binary search algorithm.

1. **If you do not initialize an array what will happen? Ans-**Array will have default value.
2. **Can we declare array size as a negative number?** No. We cannot declare the negative integer as an array size. If we declare, there will be no compile-time error.

However, we will get NegativeArraySizeException at run time.

**26) Can we use Generics with the array?**

No, we cannot use Generic with an array.

27) **What will be the output of below code?**

int myArr[] = new int [7];

System.out.print(myArr);

Ans: Output will be Garbage value.

myArr is an array variable, myArr is pointing to an array if it is integers.

Printing myArr will print garbage value. It is not same as printing myArr[0].

28) **What is the step to access elements of an array in Java?**

We can access using “index”.

Index starts from Zero(0), so the first element is stored in location zero and the last element will be Array.length – 1.

Example:-String strArr[] = new String []{“A”, “B”, “C”, “D”, “E”};

strArr[0] means “A” and strArr[2] means “C”.

* rics feature.

**Loop in Java**

**Q1: What is the syntax for " For Each loop / Enhanced Loop "?**

**Ans:** When we do not know the size or length (condition), then we should use For Each Loop:

int a[]={4,6,5,9,8};

for(int x:a)

{

}

**Q2. What is break and continue keyword in loop?**

**Ans:** A break statement results in the termination of the   statement to which  it applies (switch, for, do, or   while).   
  A continue statement is used to end   the current loop   iteration and return    control to the loop statement.

**Q.3 Can a for statement loop indefinitely?**  
  
   Ans: Yes, a for statement can loop indefinitely. For    example, consider the following: for(;;);

**String**

**1)What are Strings ?** String is a most frequently used non primitive data type which is used to store different string values. It is also immutable **Object in java.** Every immutable object in Java is thread safe ,that implies String is also thread safe. String is stored in the Constant String Pool of the heap memory.

**2) Features – a)** String is immutable and final in java which means a constant and cannot be changed once created.it can not be changed/manipulated. Manipulation for the string we cannot do.b) String is stored in the Constant String Pool.c) String is a Character type Array.d) **String** can not be used bytwo **threads** simultaneously. e))To make string we can use three ways/ class - 1) String class 2) String buffer class and 3 ) String builder class.

3) **Why use Strings? / Why do we use string in Java**?

String is used to create and manipulate character string object that value cannot be changed (read -only and immutable).b) **It** is widely **used** in **Java** programming, are a sequence of characters. C) It is used for storing and processing language is going to be the String class.

1. **What is immutable object?** Immutable object are those objects whose value can not be modified after creation. For Example -Objects of String and all the Wrapper classes.
2. **What is thread safe?** Thread safe means synchronization-When multiple threads they are going to access the same single object they will need synchronization.
3. **What Is String, StringBuffer VS StringBuilder?**

**String** is immutable sequence of characters ( once created can not be changed )or object . The object created as a String is stored in the Constant String Pool. Every immutable object in Java is thread safe ,that implies String is also thread safe . String can not be used by two threads simultaneous.

**StringBuffer- is a** thread-safe, mutable sequence of characters. It is like a [String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html), but it can be modified because of it’s mutable character .It represents growable and writeable character sequences **.It is builting and by default synchronized.**

**StringBuilder- is** a mutable sequence of characters and it is identical StringBuffer except for one important difference is that It is not synchronized, which means it is not thread safe.

**7. What are the immutable class in java?** a) Wrapper classes ( Integer.Short,Long,Double..) b) String

c) Locate d) URL and URI e) BigInteger and BigDecimal.

**8) What is String pool?**String pool is an area in java heap memory and is a pool of strings.

**9)How we can create String?**  String can be created using new keyword similar to other objects and also by assigning content in a double quotes. String str1= new String("java"); String str2="java";

10) **Final VS Immutability – 1**)By declaring the ref variable as final, we will not (won’t)get Immutability nature, because final is difference from the immutability . Immutability related to object and Final is related to reference variable.

**11) What is Final VS Finally Vs finalize?**

**Final keyword** is used to define the constant values. Constant value means that the value can’t change or modify . It is used to prevent inheritance. So to avoid the inheritance we use final keyword.

**Finally is a block**. The property of finally keyword is that whatever you have written inside the finally, it will be executed. Finally block will be executed after try catch block but before the control transfer back to its origin. It means whatever the exception is coming ,it doesn’t matter. It’s a reserve keyword.

**Finalize –** Finalize is something related to garbage collector. Like inside the memory there are lot of garbage is there, lot of objects are there. It doesn’t have any reference. If we declare that particular method as a finalize before the garbage collector, it will start cleaning up the process. So It is used to perform the clean up of memory processing.

**12) String is** immutable ( once created can not be changed )object. The object created as a String is stored in the Constant String Pool. Immutable objects are always thread-safe, that implies String is also thread safe  but their references may not be. To make their references thread-safe, we may need to access them from synchronized blocks/methods. Immutable objects are thread safe, because an immutable object is an object that is no longer modified once it has been constructed and String can not be used by two threads simultaneously.

**13) Why string is immutable or Final in java ?**

The **string is Immutable** in **Java** because **String** objects are cached in **String** pool. ... At the same time, **String** was made **final** so that no one can compromise invariant/change of **String** class e.g.- **Immutability**, Caching, hashcode calculation etc by extending and overriding behaviors.

**14) What is the difference between String and StringBuffer?**

String is immutable and StringBuffer is mutable. 2) In string class Equals methods based on contant comparison. But StringBuffer based on reference comparison, because in stringbuffer equals methods is not overridden.

**15) Final VS Immutability – 1**) By declaring the ref variable as final, we will not (won’t)get Immutability nature, because final is difference from the immutability . Immutability related to object and Final is related to reference variable.

**16)Advantages of the String Implementation in JAVA./Advantages of the String Implementation in JAVA.**

1. Compilation creates unique strings. At compile time, strings are resolved as far as possible.

Because String objects are immutable, a substring operation doesn’t need to copy the entire underlying sequence of characters.

1. Strings are implemented in the JDK as an internal char array with index offsets (actually a start offset and a character count).
2. Strings have strong support for internationalization. It would take a large effort to reproduce the internationalization support for an alternative class.

**The Disadvantages of the String Implementation are:**

The major disadvantage of String class is, you can't modify it, with in the same location. It generates too many objects in heap and wastes your memory if you try to modify it too many times. So if you want to do many modifications to your string, it is not possible to use strings.

**17)What is the difference between String created using new keyword and by assigning content in a double quotes?**

Any String created using double quotes are created in String pool and if string with same content is available in pool, it will use that. In case of String created with new keyword, String is not stored in a String pool.

**18)** **How do you reverse a word in a string in Java?**

Ans --We can **reverse each word of a string** by the help of **reverse**(), split() and substring() methods. By using **reverse**() method of StringBuilder class, we can**reverse** given **string**. By the help of split("\\s") method, we can get all **words** in an array. To get the first character, we can use substring() or charAt() method.

 19) **What is intern() method of String class?**

intern() method of String always returns a String from a String pool if it is available and will create a new one if not available.

**20) What is difference between StringBuffer and StringBuilder? StringBuffer** is thread safe because its methods are synchronized whereas StringBuilder is not thread safe.

**21) Does String provide any method to get a bytes?** Yes, we can call getBytes() method.

**22)**. **Is it possible to create our own immutable class or not?** Yes We can create

Incase of Equals Methods .

**23)** **Does String provide any method to get a bytes?** Yes, we can call getBytes() method

**24**)String class is basically implementing few interfaces- a) Serializable b) Comparable and c) CharSequence.

**a) Serializable b) Comparable and c) CharSequence ?**

**Serializable** -is a marker interface that contains no data member or method. It is used to “mark” the Java classes so that objects of these classes may get a specific capability.

**Comparable interface** is used for ordering the objects of any user-defined class. This interface is found in **java.lang.package** andcontains only one method named **compareTo(Objects)** .

**A CharSequence interface** is a readable sequence of characters. This interface provides uniform, read-only access to various kind of character sequences. It is basically parent to three classed .They are a) **String b) StringBuffer c) StringBuilder**

**25)String Methods**—

|  |  |
| --- | --- |
| **boolean equals(object another)** | Checks the equality of string with the given object |
| equalsIgnoreCase() | Compare another string without matching the case |
| Length() | Returns a string length |
| charAt(i) | Returns a character at a index ‘i’ |
| toUppercase() | Return the string in uppercase |
| toLowerCase() | Return the string in lowercase |
| replace(oldval, newVal) | Replaces all occurrences of the specified char value with the given value |
| trim() | Removes the white spaces from the beginning and ending of string |
| Contains(“value”) | Checks for the matching sequence of char value and returns true /false |
| toCharArray() | Converts a string to a new character array |
| IsEmpty() | Checks whether the string is empty or not |
| endsWith() | Checks if the string ends with the specified suffix |
| Concat() | Concatenates two strings |
| **toString() method** | returns the string representation of the object. |
| **parseUnsignedLong()** method | to return the unsigned long which is represented by the subsequence in the specified radix. |
| parseLong() method | to return the signed long which is represented by the specified radix. |

## **Java Collection**

**Q1) What is Collection and Collection framework ? When we should go for collection?**

**Collection Framework—** is a container object (group of individual objects) that holds a group of object.It defines several classes and interfaces which can be used a group of objects as single entity and it can be used to represent collection.

**Collection**- is a group of individual object as a single entity. **If we want to represent a group of individual objects as a single entity then we should go for collection.** Example -Collection of books, Collection of

Student. **It is used to store**, retrieve, manipulate (add, remove,replace) and communicate aggregate data.

*There are three types of collection are supported the collection framework.-Set, List and Map.*

**Collection Interfaces-**defines the most common methods which are applicable for any collection object. In general collection interfaces is considered as root interface of collection framework. Interface provides more information than classes.Interface method- like 1) **Get method** to get an object. 1) Remove method- to remove an object 3) IsEmpty method- is the collection empty or not. Collection is the root interface of collection framework. Collection interface doesn’t not contain any method to retrieve objects, there is no concrete class which implements collection class directly.

**Q2) Difference between Collection and Collections**?

**Collection** is an interface which can be used to represent a group of individual objects as a single entity. Example -Collection of books, Collection of Student.

**Collections –** is an utility class present in Java.util.package to define several utility methods {Like sorting, Searching} for collection objects.

**Q) Nine key interfaces of collection framework ---**

**Collection interface,**

List interface List Interface Set Interface Queue Map In

Sorted set Navigable set Sorted Navigable

**Q3) Why need to go Collection and Collection Framework?**

**Collection –**Individual variable approach is the first level. There are some problem are there. If we want to represent huge number of values , declaring those many number of variable never be a good programming practice. To overcome this problem we should go for Arrays. In Array’s there are end number of problem like -Fixed in size, it can hold only homogeneous data and there is no underlying data structure, that’s why readymade method support we can’t expect. So to overcome this problem we should for collection.

**Q4) List Interface-** is the child interface of Collection. If we want to represent a group of individual objects as a single entity **where duplicates are allowed and insertion order must be preserved** then we should go for List interface. There are various implementation classes are in List interface-1) Array List (1.2), 2) Linked List(1.2), 3) Vector (1.0) and 4) Stack (1.0). Collection came in 1.2 version.

Collection came from 1.2 version. Vector and Stack concept came in 1.0 version. These two classes came from old generation. That’s why these classes also known as Legacy class.

**Collection Page 2**

**Q7) Set Interface-** **-** is the child interface of Collection. If we want to represent a group of individual objects as a single entity **where duplicates are not allowed and insertion order not preserved** then we should go for Set interface. The implementation classes for the set interface-1) Hash Set (1.2), 2) Linked HashSet(1.4) 3) Tree set(). Tree class is under the navigable interface. There are two Child interface for the set interface- 1) Sorted Interface and 2) Navigable interface.

**Q 8) What is Sorted Interface and Navigable Set?**

**Sorted Set**  -It the child interface of Set. **If we want to represent a group of individual objects as a single entity where duplicates are not allowed and but all objects should be inserted according to the some sorting order** then we should go for Sorted Set interface.

**Navigable Set-**-It the child interface of Sorted Set which defines several methods for navigation purpose.. **If we want to represent a group of individual objects as a single entity where duplicates are not allowed and but all objects should be inserted according to the some sorting order** then we should go for Sorted Set interface.

Collection(1.2)-> Set (1.2)-> Sorted Set(1.2)->Navigable Set (1.6)🡪 Tree Set(1.2)

**Q9) What is the implementation class of Navigable set?—**Tree set is the implementation class of Navigable set.

**Q10) How to Convert Array to Array List?**

**Array’s can be created from both primitives and object while Arraylist can only be created from objects.**

Syntax for object Array to ArrayList---- List<T> arrayList = Arrays.asList(<arrayname>);

Example- String[]names={“ Ruma”, “Afsheen”,”Taseen”}; List<String> nameList = Arrays.asList(<name>);

**11) What is Queue?** It is the child interface of collection. .. **If we want to represent a group of individual objects prior to processing then** we should go for queue. **Example- Before** sending a mail all mail Id’s we have to store somewhere and in which order we saves, in the same order mail’s should be delivered.[First in First out] . For this requirement Queue concept is the best choice.

**Collection (**1.2)🡪 Queue (1.5)🡪Priority Queue(1.5)🡪 Blocking Queue(1.5) then Blocking Queue🡪 Linked B.Q(1.5) and Priority B.Q(1.5).

**12) Map Interface -- Map interface** is not the child interface of collection. If we want to represent a group of individual objects **as key value pairs** then we should go for Map interface. Both key and value are objects, keys are not duplicated but values can be duplicated. **The Implementation classes of Map Interface:--**

Map interface-🡪1) HashMap(1.2)🡪Weak HashMap(1.2)🡪 Identify HashMap(1.4)🡪 Hash Table(1.0)

Hash Map 🡪 Linked hash Map (1.4) And Hash Table 🡪 Properties (1.0).

HashTable is the child of Dictionary which is abstract method. Properties is the child of HashTable.

**Legacy Classes –**Dictionary, HashTable and Properties , these three things came from 1.0 version. That’s why these three things are considered as **Legacy Classes**.

**The Child Interface of Map Interface**-

**1) Sorted Map** –It is the child interface of Map interface. If we want to represent a group of Key Value pairs according to some sorting order of keys then we should go for Sorted Map.

1. **Navigable Map**- It is the child interface of sorted Map Interface. It defines several utility methods for navigation purpose. Map (1.2)🡪 Sorted Map(1.0),🡪 Nqvigable Map(1.6),🡪 Tree Map(1.2).

**Note Tree Set and Tree Map- Similarities** ---In total collection framework Heterogeneous objects are allowed except Tree set and tree Map. In TreeSet and TreeMap, the objects are going to be stored in some sorting order. Tree set means Sorted Set, Some sorting order and Tree Map also Means some sorting order.

## **Array List –** is the class of List interface. It is implemented based on Data structure. That’s why it is Resizable or Growable array. Duplicate elements are allowed, Insertion order is preserved. Heterogeneous object is possible, happily we can insert. Array List is best choice if our frequent operation is retrieval operation. Because ArrayList Implements Random Access Interfaces.

**By default ArrayList** is NonSynchronized but we can get synchronized version of ArrayList by using Collection class **Synchronized List() method**.

**ArrayList Constructor –**

1. ArrayList al = new ArrayList(); // Creates an Empty ArrayList with default initial capacity 10.
2. ArrayList al = new ArayList(Int initial capacity);
3. ArrayList al = new ArrayList(Collection c);

**Iterator-** **It is not bidirectional Cursor because by using Iterator we can move only to the forward direction. It is universal cursor because we can apply this concept for any collection objects. Replacement of new objects perform is not possible only read and remove operation is possible.**

…………………………………..

**Casting in Java**

Casting is the temporary conversation of a variable from its original data type to some other data type.  If a class shares an IS-A or inheritance relationship with another class or interface, their variables can changing value from one data type to another data type is know as data type conversation.

There two types of casting—1) Up casting 2) Down casting.

**Up Casting-** is a dynamic polymorphism. It is done by creating Child class object can be referred by parent class ref variable. Parent class p = new Child();

**Down casting –** – it is not allowed at run time .parent p = new child(); then child c = (child)p; or Child class cl = (child class) new parent class (); so down casting is possible. We have to convert into child class like this Child class cl = (child class) new parent class (); using child class reference variable and write the name of the child class just before new keyword within bracket then parent class object. but it will give an Exception “ClasscastException”.

**Exception Handling**

**What is Exception and Exception Handling in Java?**

**Exception -**is a run time error. It is an abnormal condition that arises in a code sequence at run time.

**Exception Handing-** it is one of the powerful mechanism to handle the run time error. It is managed by 5 keywords. 1) Try 2) Catch 3) finally 4) throw 5) throws.