Topic - Fundamental

B1). Feature of java. (or) java buzzword.

1. Simple
2. Object-Oriented
3. Portable
4. Platform Independent
5. Secured
6. Robust
7. Architecture Neutral
8. Interpreted
9. High Threaded
10. Multithreaded
11. Distributed
12. Dynamic

B2). Difference between JDK, JRE and JVM.

* JDK : Java Development Kit
* The JDK is a development environment for building applications, applets and component using the java programming language.
* Development tools such as a Compiler, Debugger etc.
* JDK provides the environment to develop and execute java source code.
* JDK is platform dependent.
* JRE : Java Runtime Environment
* JRE is a piece of a software which is designed to run other software.
* It contains the Class libraries, Loader class and JVM.
* If you want to run java program you need JRE.
* Cannot write and compile the java program.
* JRE has JVM.
* JRE is platform dependent.
* JVM : Java Virtual Machine.
* JVM converts java byte code into machine language.
* JVM is part of JRE.
* JVM is an environment for executing byte code.
* JVM provides a platform-independent way of executing java source code.
* Once you run java program, you can run on any platform and save lots of time.
* JVM is platform dependent.

B3). Java is platform independent?

* Yes.
* java is platform independent because it is different from other language like c, c++, etc. which are complied into platform specific machine while java is wrote once, run anywhere language . a platform is the hardware or software environment in which a program runs. java code can be executed on multiple platforms, for example, windows, Linux, sun Solaris , Mac/os etc. java code is compiled by the complier and converted into byte code. this byte code is a platform-independent because it can be run on multiple platform, i.e., write once run anywhere(wora).

B4). Three flavors of Java?

* Program are written in three basic flavors:
* 1) Applets: It is run in the JVM built into a web browser.
* 2) Applications: It is run in the JVM installed once in a computer system.
* 3) Servlets / JSP pages: It is run in the JVM installed on a web server.

B5). How many types of memory areas are allocated by JVM?

* JVM has five memory locations
* 1). Heap
* 2). Stack
* 3). PC Register
* 4). Execution Engine
* 5). Native Method Stacks

B6). What is the latest version of java?

* The latest version of java is JAVA-17 or JDK17.

B7). What is Write Once, Run Anywhere (WORA)?

* Java code is compile by the compiler and converted into byte code. This byte code is platform – independent because it can be run on multiple platform etc.. Window, Linux, Mac/os.

B8). Is java a pure object oriented language?

* No.
* It support primitive data type like int, float, double etc. Which are not object. Because in java we use data types like int, float, double etc which are not object oriented. That is why java is not 100% object oriented.

B9). What is Byte code?

* Java byte code is the instruction set for the Java Virtual Machine.
* With the help of byte code we achieve platform independent in java.
* When we write a program in java, Firstly, the compiler compile that program and generate a byte code.

B10). What is Heap space in java?

* The java heap is the area of memory used to store objects instantiated by applications running on the JVM.

B11). Difference between EAR, JAR and WAR file in J2EE.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. | JAR | WAR | EAR |
| 1 | A jar file is a file with java classes, associated metadata, and resources such as text and images aggregated in one file. | A war file is a file that is used to distribute a collection of JAR files, JSP, servlet, XML files, static web pages like HTML and other resources that constitute a web application. | An ear file is a standard jar file that represent the modules of the application, and metadata directory called META-INT which contains one or more deployment description. |
| 2 | Stands for java archive. | Stands for Web Application Resource or Web Application Archive. | Stands for Enterprise Application Archive. |
| 3 | Has.jar file extension. | Has.war file extension | Has.ear file extension |
| 4 | Allows JRE to deploy an entire application including the classes and related resources in a single request. | Allow testing and deploying web applications easily. | Allow deploying different modules onto an application server simultaneously. |

B12). Explain memory leak in java.

* A memory leak is a situation when unused objects occupy unnecessary space in memory.
* A memory leak is a situation when there are objects presents in the heap that are no longer used, but the garbage collector is unable to remove them from memory and, thus are unnecessarily maintain.
* It is bad because it blocks memory resources and degrades system performance over time.
* OutOfMemoryError heap error in the application.

B13). How garbage collections work in java?

* Garbage means unreferenced objects.
* Garbage collection is process of destroying the unreferenced objects.
* In java, garbage collection is the process of managing memory, automatically.
* It finds the unused objects and delete or remove them to free up memory.
* The garbage collection mechanism uses several GC algorithms.

B14). Does java garbage collector clean both heap and stack memory?

* Garbage collector in java works only on Heap memory not on Stack memory. Because of the main principal that stacks works on Last In First Out.

B15). Why garbage collection is required in java?

* It makes java memory efficient because garbage collector removes the unreferenced objects from heap memory.

Topic – Array

B1). How do I initialize a String array?

* String [ ] str = new String{“yatin”, “sam”, “jay”};

B2). What is array?

* Array is a collection of similar data elements.

B3). How many types of array available?

* Two types of array
* 1). Single dimensional array
* 2). Multi dimensional array : data is store in row and column based index.
* Syntax to declare array

- data Type[] arr;

- data Type []arr;

- dataType arr[];

- Syntax to declare and initialize array

- data type arr[] = new data type[size];

B4). Is array are considered as primitive data type?

* No.
* Array is consider to be an object in java.
* Array can be created using the “new” keyword. That’s reason array is not primitive data type.

B5). How do I create a list from array which is completely independent of the original array?

* Using “ Arrays.asList(array name) “ method.

B6). What is the index of the first element in an array?

* 0(zero).

B7). How do you print the content of an array in java?

* We cannot print array elements directly in java.
* One-dimensional array : Arrays.toString(array name).
* Multi-dimensional array : Arrays.deepToString(array name).

B8). How do you print the content of a multi-dimensional array in java?

* Multi-dimensional array : Arrays.deepToString(array name);

B9). Why is it good practice to store sensitive information like password, SSN into a character Array rather than string?

* String is immutable, there is no methods define to change or overwrite the content of the string.
* An immutable feature makes string objects unstable for storing sensitive information like password, SSN etc.
* We should always store the secure information on char [] rather than string.

B10). Which algorithm does arrays? Sort use in java?

* It uses a stable, adaptive, iterative implementation of merge sort algorithm for array of object.

Topic – String

B1). Can string be referred as a data type?

* Yes.
* A string in java is actually a non-primitive data type, because it refers to an object.

B2). What is toString() method?

* If you want to represent any object as a string, toString() method comes into existence.
* The toString() method returns the string representation of the object.

B3). All the string objects created using string literals are stored in string pool?

* Yes
* String pool is a storage area in heap memory.
* Where string literals store, it is also know string intern pool.
* All string are store in the string pool, that is allocated in the java heap.
* 1). String str1 = “yatin”; - save in string pool.
* 2). String str2 = new String (“Yatin”).intern (); - save in string pool.
* 3). String str2 = new String (“Yatin”); - save in heap memory.

B4). Difference between StringBuffer and StringBuilder?

* Java provides three classes to represent a sequence of characters: String, StringBuffer, StringNuilder.
* The string class is an immutable class but Stringbuffer and Stringbuilder are mutable classes.

|  |  |
| --- | --- |
| * StringBuffer | * StringBuider |
| * 1). It is synchronized. | * 1). It is non-synchronized. |
|  |  |
| * 2). Thread safe. It means two threads can’t call the methods of StringBuffer simultaneously. | * 2). Not thread safe. It means two threads can call the method of StringBuilder simultaneously. |
|  |  |
| * 3). StringBuffer is less efficient than StringBuilder. | * 3). StringBuilder is more efficient than StringBuffer. |
|  |  |
| * 4). StringBuffer was introduced in java 1.0. | * 4). StringBuilder was introduce in java 1.5. |

B5). What is immutable?

* Immutable means unmodifiable.
* Strings in java are immutable.
* Original content of string is never changed.

B6). Is string is immutable?

* Yes.
* String is immutable (once created cannot be change) object.
* String object is stored in the Constant String Pool.
* Every immutable object in java is thread safe.

B7). How do you create a string object?

* There are two ways to create a string object:
* 1). By string literal: String str = “Hello”;
* Java String literal is created by using double quotes(“ ”).
* 2). By new keyword: String s = new String(“Hello”);
* Java String is created by using keyword “new”.

B8). What is the difference between creating string object using new keyword and string literals?

* When you create a string object by using “new” keyword, It always create a new object in heap memory.
* When you create a string object by using string literals, It create a object in string pool.

B9). What is java string pool?

* String pool is nothing but a storage area in java heap where string literals stores.
* String pool also knows as String Intern Pool or String Constant Pool.
* By default, it is empty and privately maintained by the java string class.
* Creating a number of string may increase the cost and memory too which reduce the performance also.

B10). Immutable objects are thread safe?

* Yes.

B11). Overriding toString() example.

* Answer in B11 program.

B12). What is string interpolation in java?

* String interpolation is a process in which the placeholder characters are replace with variable (or strings ) which allows to dynamically or efficiently print out text output.
* String interpolation make code more compact
* These variables are then replaced with their actual value at runtime.
* Method for perform string interpolation.
* 1. Using the ‘+’ operator
* 2. Using format() function
* 3. Using Message Format class
* 4. Using StringBuilder class.

B13). Name the interfaces that java string class implements.

* The java.lang.String class implements Serializable, Comparable and CharSequence interfaces.

B14). How do I compare strings in java?

* Three ways to compare strings in java
* 1). By using equals() method.
* 2). By using == operator.
* 3). Y using compareTo() method.

B15). Can we have case null in string switch case?

* The prohibition against using null as a switch label prevents one from writing code that can never be executed. If the switch expression is of a reference type, that is, String or a boxed primitive type or an enum type, than a run time error will occur if the expression evaluates to null at run time.

B16). What is the default implementation of equals() method in object class?

* The default implementation of equal() method is defined in java.lang.Object class, which simply checks if two object references (say x and y) refer to the same object. It is also known as shallow comparison.

Topic – OOPs Concept

B1). What is inheritance?

* Inheritance is a mechanism in which one object receives all the properties and behaviors of a parent object. It is an important part of OOPs.
* The property of super class extends into sub class.
* The idea behind inheritance in java is that you can create new classes.
* Inheritances represent parent child relationship.
* Main purpose is: Reusability and extensibility of code.
* “extends” keyword through generate inheritance.
* Always last child class to create object and access whole properties of parent class except private.
* There are 5 types of inheritance
* 1). Single 2). Multilevel 3). Hierarchical
* 4). Multiple 5). Hybrid

B2). Which inheritance is not supported by java? Why?

* Multiple and Hybrid inheritance does not support directly.
* Java does not support multiple inheritances to avoid the ambiguity caused by the multiple inheritances. One of the most common problems that are occurred due to multiple inheritances is the diamond problem.

B3). What is advantage of inheritance?

* The biggest advantage of inheritance is code reusability.
* “Extensibility”, you can add new features or change the existing features easily in sub-class.
* We can achieve runtime polymorphism (method overriding).
* Inheritance make easy to maintain code.
* It also helps to reduce code duplicity.

B4). Difference between inheritance and encapsulation.

|  |  |  |
| --- | --- | --- |
| Sr. | Inheritance | Encapsulation |
| 1 | Inheritance is a mechanism in which one object receives all the properties and behaviors of a parent object. | Encapsulation is a wrapping up of data into single unit. |
| 2 | Inheritance indicates that a child class inherits all the attributes and methods from a parent class. | Encapsulation indicates that one class must not have access to the data of another class. |

B5). Difference between inheritance and abstraction.

|  |  |  |
| --- | --- | --- |
| Sr. | Inheritance | Abstraction |
| 1 | Inheritance allows using properties and methods of an already existing class. | Abstraction allows hiding the internal details and displaying only functionality to the users. |
| 2 | Inheritances help to improve code reusability. | Abstraction help to reduce and complexity of the code. |

B6). Difference between inheritance and polymorphism.

|  |  |  |
| --- | --- | --- |
| Sr. | Inheritance | Polymorphism |
| 1 | Inheritance is a mechanism in which one object receives all the properties and behaviors of a parent object. | Ability to take one name having different forms.  Many forms. |
| 2 | It is basically applied to classes. | It is basically applied to functions or methods. |
| 3 | Inheritance supports the concept of reusability and reduces code length. | Polymorphism allows the object to decide which form of the function to implement at compile time (overloading) as well as run time (overriding). |
| 4 | Inheritance can be single, hybrid, multiple, multilevel, hierarchical. | Polymorphism can be compiled-time (overload) as well as run-time (overriding). |
| 5 | It is used in pattern designing. | It is also used in pattern designing. |

B7). Can we override static method in java?

* No.
* We can declare static method with the same signature in the subclass, but it is not consider method overriding as there won’t be any run-time polymorphism.

B8). Can we overload static method in java?

* Yes.
* We can have two or more static methods with the same name, but differences in input parameter.

B9). Can a class implement more than one interface?

* Yes.
* So the implements keyword is followed by a comma-separated by list of the interfaces implemented by the class.
* A class implements multiple interfaces.

B10). Can a class extend more than one class in java?

* No.
* A class can only extend a single class.
* A class extends two classes, that is multiple inheritance, java does not support multiple inheritance directly, so can a class extend more than one class in java is not possible.

B11). Can an interface extend more than one interface in java?

* Yes.
* An interface can extend multiple interfaces.

B12). What will happen if a class implements two interfaces and they both have a method with same name and signature?

* This is the general rule of inheritance, method overriding, hiding and declarations and applies also to possible conflicts not only between 2 inherited interfaces method.

B13). Can we pass an object of a subclass to method expecting an object of the super class?

* Yes, you can pass that because sub class and super class are related to each other by inheritance which provides IS-A property.

B14). Are static member inherited to sub classes?

* Yes, static member also inherited into subclass in java.

# B15). What happen if the parent and the child class have a field with same identifier?

* Sub class field will hide the super class field. Hidden super class field in sub class can be accessed using “super” keyword.

# B16). Are constructors and initializes also inherited to sub class?

* No.
* Constructers are not members, so they are not inherited by subclasses, but the constructer of the super class can be invoked from the sub class.

# B17). How do you restrict a member of a class from inheriting by its sub class?

* We can restrict a member of a class from inheriting to its sub classes by declaring the member as a private.
* Because, private member are not inherited to sub classes.

# B18). How do you implement multiple inheritances in java?

* The only way to implement multiple inheritances is to implement multiple interfaces in a class.
* In java one class implements two or more interfaces.

B19). Can a class extend itself in java?

* No.

B20). How do you override a private method in java?

* No.
* The program gives a compile time error.
* Answer in code.

B21). When to overload a method in java and when to override it?

* Method overload: If a class has multiple methods having same name but different in parameters, it is knows as method overloading.
* If we have to perform only one operation, having same name of the methods increases the readability of the program.
* F
* Method overriding: If sub class has the same method as declared in the parent class, it is known as method overriding in java.
* Method overriding is used to provide the specific implementation of a method which is already provides by its super class.

B22). What the order is of extends and implements keyword on java class declaration?

* The “extends” always precedes the “implements” keyword in any java class declaration.

B23). How do you prevent overriding a java method without using the final modifier?

* Static methods cannot be overridden.
* We cannot override the static methods in a derived class because static methods are linked with the class.
* Private method cannot be overriding.
* Private methods of the parent class cannot visible in child class, hence they cannot be overridden.

B24). What are the rules of method overriding in java?

* Both the super class and sub class must have the same method name, same return type, same parameter list.
* We cannot override the method declare as “final” and “static”.

B25). Difference between method overloading and method overriding in java.

|  |  |  |
| --- | --- | --- |
| Sr. | Method Overloading | Method Overriding |
| 1 | It occurs within the class. | It is performed in two classes with inheritance relationships. |
| 2 | Method overloading is compile time polymorphism. | Method overriding is run-time polymorphism. |
| 3 | It helps to increase the readability of the program. | It is used to the specific implementation of the method which is already provided by its parent class or super class. |
| 4 | Method overloading may or may not require inheritance. | Method overriding always needs inheritance. |
| 5 | In method overloading, methods must have the same name and different signature. | In method overriding, methods must have the same name and same signature. |
| 6 | In method overloading, the return type can or cannot be the same. | In method overriding, the return type must be the same. |

B26). What happens when a class implements two interfaces and both declare field (variable) with same name?

* This is the general rule of inheritance, method overriding, hiding and declarations and applies also to possible conflicts not only between 2 inherited interfaces method.

B27). Can a subclass instance method override a super class static method?

* No, we cannot override static method because method overriding is based on dynamic binding at run time and the static methods are bonded using static binding at compile time.

B28). Can a sub class static method hide super class instance method?

* No, it results in compilation error in the sub class.

B29). Can a super class access sub class member?

* No, because a super class has no knowledge of its sub classes.

B30). Difference between object oriented and object based language?

|  |  |  |
| --- | --- | --- |
| Sr. | Object Oriented | Object Based |
| 1 | They support built-in objects. | They do not support built-in objects. |
| 2 | They support all the features the OOPs including inheritance and polymorphism. | They support the usage of object and encapsulation. They do not support inheritance or polymorphism. |
| 3 | C#, Java, VB.Net | JavaScript, VB |

B31). Explain diamond problem.

* The diamond problem is related to multiple inheritances.
* Sometimes it is also known as the deadly diamond problem or deadly diamond death.
* Java does not allow is multiple inheritances where one class can inherit properties from more than one class. It is known as the diamond problem.

B32). Why java does not support operator overloading?

* Simple and Clear design is one of the goals of the java designer.
* Avoid programming errors: operator overloading them will come up with multiple meanings for same operator which will make the learning hard and things more confusing.
* JVM complexity
* Easy development of tools.

B33). What is encapsulation in java?

* Encapsulation is wrapping up of data into single unit.
* We can create a fully encapsulated class in java by making all the data members of the class private.
* Now we can use getter methods to set and get the data in it.
* The Java Bean is encapsulated class.

B34). Which of the java OOPS feature promotes access protection or data hiding?

* Encapsulation promotes access protection and data hiding.
* Abstraction is used for hiding the unwanted data and giving relevant data
* Encapsulation means hiding the code and data into single unit to protect the data from outside world.

Topic – Exception

# B1). How the exceptions are handling in java?

* The “try – catch” is the simplest method of handling exceptions. Put the code you want to run in the “try” block, and any java exceptions that the code throws are caught by one or more “catch” block.
* You can’t use a “try” block alone. The “try” block should be immediately followed either by a “catch” or “finally” block.

# B2). Difference between error and exception in java.

|  |  |  |
| --- | --- | --- |
| Sr. | Error | Exception |
| 1 | All error in java are unchecked type. | All exception in java are checked type and unchecked type. |
| 2 | It is irrecoverable. | It is recoverable. |
| 3 | It can’t be occur at compile time. | It can be occur at run time and compile time. |
| 4 | It belongs to java.lang.error. | It beongs to java.lang.Exception. |
| 5 | OutOfMemoryError, IOError | NullPointerException, SqlException |

# B3). What are checked and unchecked exceptions?

* 1). Checked exception : The exceptions that are checked during the compile time are termed as Checked exception in java. Ex, SQLException, IOException, ClassNotFoundException, FileNotFoundException etc.
* 2). Unchecked Exception : An exception that occurs during the execution of a program is called an unchecked exception or runtime exception.
* The compiler generally ignores the unchecked exceptions during compilation. Unchecked exceptions are checked during the runtime. Ex, ArithmeticException, NullPointerException, NumberFormatException.

# B4). How do we handle more than one type of exception using catch block in java?

* It is possible for a single catch block to catch multiple exceptions by separating each with | (pipe symbol) in the catch block.
* catch (Exception type 1 | Exception type 2).

B5). What happens if an exception is thrown from the finally or catch block in java?

* That exception propagates out and up, and will be handled at a higher level. Your finally block will not be completed beyond the point where the exception is thrown.

B6). Will the finally block be executed when the catch clause throws exception in java?

* No matter whether an exception occur in try-block or not, finally will always executed.

B7). What are a user defined / custom exception in java?

* User defined exception or custom exception is creating your own exception class and throws that exception using ‘throw’ keyword.

B8). Does a finally block always run in java?

* Yes, the finally block is always get executed.

B9). Does return statement allow finally block to execute in java?

* Yes, the finally block will be executed even after a return statement.

B10). Should a catch block always follow try block in java for exception handling?

* A catch block must follow the try block.

B11). Difference between error and runtime exception in java?

* An error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch.
* Runtime exception is the super class of those exceptions that can be thrown during the normal operation of the java virtual machine.

B12). Difference between throw and throws clause in java?

|  |  |  |
| --- | --- | --- |
| Sr. | throw | throws |
| 1 | The throw keyword used inside a function. It is used when it is required to throw an exception logically. | The throws keyword is used in the function signature. It is used when the function has some statements that can lead to exceptions. |
| 2 | The throw keyword is used to throw an exception explicitly. It can throw only one exception at a time. | The throws keyword can be used to declare multiple exceptions, separated by a comma. Whichever exception occurs, if matches with the declared ones, is thrown automatically then. |
| 3 | Syntax of throw keyword includes the instance of the exception to be thrown. Syntax wise throw keyword is followed by the instance variable. | Syntax of throws keyword includes the class names of the exceptions to be thrown. Syntax wise throws keyword is follow by exception class name. |

B13). Can try block exist without any catch and finally block in java?

* No, we cannot exist only try block without catch and finally block.

B14). What is stack trace?

* A stack trace is a list of the method calls that the application was in the middle of when an exception was thrown.

B15). What is the order of catch blocks when catching more than one exception?

* At a time only one exception occurs and at a time only one catch block is executed.
* All catch block must be ordered from most specific to most general, i.e. catch for ArithmeticException must come before catch for Exception.

B16). Can we use FileNotFoundException and IOException in java multi catch?

* No, you will get compile time exception, The exception FileNotFountException is already caught by the alternative IOException.

B17). Give the few examples of checked exceptions.

* ClassNotFoundException, IOException, SQLException.

B18). Give the few examples of unchecked exceptions.

* ArithmeticException, NullPointerException, InputMismatchException.

B19). Explain exception handling when overriding a method?

* If the superclass method does not declare an exception, subclass overridden method cannot declare the check exception but it can declare unchecked exception.
* If the superclass method declares an exception, subclass overridden method can declare same, subclass exception or no exception but cannot declare parent exception.

B20). Can overridden method throw RuntimeException when original methods throw ArithmeticException?

B21). Can I write try block without any catch and finally block?

* No, we cannot write only try block without catch and finally block.

B22). Difference between final, finally and finalize in java?

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. | final | finally | finalize |
| 1 | Final is the keyword and access modifier which is used to apply restriction on a class, method and variables. | Finally is the block in java exception handling to execute the important code whether the exception occurs or not. | Finalize is the method in java which is used to perform clean up processing just before object is garbage collected. |
| 2 | Final keyword is used with the classes, variables and methods. | Finally is always related to try and catch block in exception handling. | Finalize () method is used with the objects. |
| 3 | Final method is executed only when we call it. | Finally bock is executed as soon as try and catch block is executed. | Finalize method is executed just before the object is destroyed. |

B23). What is re-throwing an exception?

* Re-throwing an exception means calling the throw statement without an exception object, inside a catch block.

B24). Explain the rules of exception handling in terms of method overriding?

* If the superclass method does not declare an exception, subclass overridden method cannot declare the check exception but it can declare unchecked exception.
* If the superclass method declares an exception, subclass overridden method can declare same, subclass exception or no exception but cannot declare parent exception.

Topic – Thread

# B4). Can we start a thread twice? TestThread t1 = new TestThread(); // t1.start. // t1.start.

* No.
* After starting a thread, it can never be started again.
* If you does so, an illegalThreadStateException is thrown.
* In such case, thread will run once but for second time, it will throw exception.

# B5). What if we call java run() method directly without calling start?

* Each thread starts in a separate call stack.
* We can see in the B5 program that there is no context switching because here t1 and t2 will be treated as normal object not thread object.
* You can not directly call run method to create a thread, you need to call start method to create a new thread.

Topic – Collection

B1). How do you initialize an array list?

* Initialization with add () method.
* Initialization with asList () method.
* Initialization with List.of () method.
* Initialization with using another collection.
* All answer in B1 program in collection assignment.

B2). What is java collections framework?

* The collection framework represents a unified architecture for storing and manipulating a group of objects.

B3). What is the difference between list and set?

|  |  |  |
| --- | --- | --- |
| Sr. | List | Set |
| 1 | List is ordered and can contain the same elements in it. | Set is unordered and contains different element in it. |
| 2 | The list implementation classes are LinkedList and ArrayList. | The set implementation classes are TreeSet, HashSet and LinkedHashSet. |
| 3 | The list implementation allows us to add the same or duplicate elements. | The set implementation does not allow us to add the same or duplicate elements. |
| 4 | The insertion order is maintained by the List. | It doesn’t maintain the insertion order of elements. |
| 5 | It is used when we want to frequently access the elements by using the index. | It is used we want to design a collection of distinct elements. |

B4). What is the difference between Map and Set?

|  |  |  |
| --- | --- | --- |
| Sr. | Set | Map |
| 1 | Set is unordered and contains different elements. | Map contain the data key – value pair. |
| 2 | Set is used to construct the mathematical set in java. | Map is used to do mapping in the database. |
| 3 | It cannot contain repeated value. | It can have same value for different key. |

B5). What are the classes that implements list and Set interfaces?

* List interface are implemented by the classes ArrayList, Vector, LinkedList, Stack.
* Set interface are implements by the classes HashSet, TreeSet, LinkedHashSet, EnumSet.

B6). What is an iterator?

* An iterator is an object that can be used to loop through collections, like ArrayList and HashSet. It is called an “iterator” because “iterating” is the technical term for looping.

B7). What is the difference between Iterator and Enumeration?

|  |  |  |
| --- | --- | --- |
| Sr. | Iterator | Enumeration |
| 1 | Iterator is a universal cursor as it is applicable for all the collection classes. | Enumeration is not a universal cursor as it applies only to legacy classes. |
| 2 | Iterator has a remove () method. | Enumeration does not have the remove () method. |
| 3 | Iterator can do modifications ( using remove () method it removes the element from the collection during traversal). | Enumeration interface acts as a read only interface, one can do not any modifications to collection while traversing the elements of the collection. |

B8). What is the difference between HashMap and Hashtable?

|  |  |  |
| --- | --- | --- |
| Sr. | HashMap | Hashtable |
| 1 | HashMap is non synchronized. | Hashtable is synchronied. |
| 2 | It is not thread safe and can’t be share between many threads without proper synchronization code. | It is thread safe and can be share with many thread. |
| 3 | HashMap allows one null key and multiple null values. | Hashtable doesn’t allow any null key or value. |
| 4 | HashMap is fast. | Hashtable is slow. |

B9). What is difference between Iterator and ListIterator.

|  |  |  |
| --- | --- | --- |
| Sr. | Iterator | ListIterator |
| 1 | Can traverse element present in collection only in the forward direction. | Can traverse elements in collection both in forward and backward direction. |
| 2 | Helps to traverse Map, List and Set. | Can only traverse List and not the other two. |
| 3 | Cannot modified or replace elements present in collection. | Can easily add elements to a collection at any time. |
| 4 | Certain methods of iterator are next(), remove() and hasNext(). | Certain method of Listiterator are next(), previous(), hasNext(), hasPrevious(), add(). |

B10). What is the difference between Array and Arraylist in java?

|  |  |  |
| --- | --- | --- |
| Sr. | Array | ArrayList |
| 1 | An array is dynamically created object. It serves as a container that holds the constant number of values of the same type. | The arraylist is a class of java collections framework. It contains popular classes like Vector, Hashmap and HashTable. |
| 2 | Array is static in size. | ArrayList is dynamic in size. |
| 3 | An array is a fixed length data structer. | ArrayList is variable length data stricter. It can be resize itself when needed. |
| 4 | An array can store both objects and primitive type. | We cannot store primitive type in arraylist. It automatically convert primitive type to object. |

B11). List the differences between LinkedList and ArrayList in java.

|  |  |  |
| --- | --- | --- |
| Sr. | ArrayList | LinkedList |
| 1 | ArrayList internally uses a dynamic array to store the elements. | Linkedlist internally uses a doudly linked list to store the elements. |
| 2 | Arraylist is better for storing and accessing data. | Linkedlist is better for manipulating data. |
| 3 | An arraylist class can act as a list only because it implements list only. | LinkedList class can act as a list and queue both because it implements list and deque interfaces. |

B12). Difference between comparable and comparator interface.

|  |  |  |
| --- | --- | --- |
| Sr. | Comparable | Comparator |
| 1 | Comparable provides a single sorting sequence. In other words, we can sort the collection on the basis of a single element such as id, name and price. | The comparator provides multiple sorting sequences. In other words, we can sort the collection on the basis of multiple elements such as id, name and price. |
| 2 | Comparable affects the original class, i.e., the actual class is modified. | Comparator doesn’t affect the original class, i.e., the actual class is not modified. |
| 3 | Comparable provides compareTo() method to sort elements. | Comparator provides compare() methods to sort elements. |
| 4 | Comparable is present in java.lang package. | A comparator is present in the java.util package. |