

1) Why pointers are eliminated from java.?

Ans) a. Pointers lead to confusion for a programmer.

b. Pointers may crash a program easily, for example, when we add two pointers, the program crashes immediately.

c. Pointers break security. Using pointers, harmful programs like Virus and other hacking programs can be developed. Because of the above reasons, pointers have been eliminated from java.

2) What is the difference between a function and a method.?

Ans). A method is a function that is written in a class. We do not have functions in java; instead we have methods. This means whenever a function is written in java. It should be written inside the class only. But if we take C++, we can write the functions inside as well as outside the class. So in C++, they are called member functions and not methods.

3) Which part of JVM will allocate the memory for a java program.?

Ans). Class loader subsystem of JVM will allocate the necessary memory needed by the java program.

4). Which algorithm is used by garbage collector to remove the unused variables or objects from memory.?

Ans). Garbage collector uses many algorithms but the most commonly used algorithm is mark and sweep.

5). How can you call the garbage collector?

Ans). Garbage collector is automatically invoked when the program is being run. It can be also called by calling gc() method of Runtime class or System class in Java.

6) What is JIT Compiler ?

Ans). JIT compiler is the part of JVM which increases the speed of execution of a Java program.

7) What is an API document ?

Ans). An API document is a .html file that contains description of all the features of a software, a product, or a technology. API document is helpful for the user to understand how to use the software or technology.

8) What is the difference between *#include* and *import* statement?

Ans). *#include* directive makes the compiler go to the C/C++ standard library and copy the code from the header files into the program. As a result, the program size increases, thus wasting memory and processor's time. *import* statement makes the JVM go to the Java standard library, execute the code there, and substitute the result into the program. Here, no code is copied and hence no waste of memory or processor's time. so import is an efficient mechanism than *#include*.

9) What is the difference between *print()* and *println()* method ?

Ans). Both methods are used to display the results on the monitor. *print()* method displays the result and then retains the cursor in the same line, next to the end of the result. *println()* displays the result and then throws the cursor to the next line.

10) What happens if String args[] is not written in main() method ?

Ans). When main() method is written without String args[] as:

```
Public static void main( )
```

The code will compile but JVM cannot run the code because it cannot recognize the main() as the method from where it should start execution of the Java program. Remember JVM always looks for main() method with string type array as parameter.

11) What is the difference between float and double?

Ans). Float can represent up to 7 digits accurately after decimal point, whereas double can represent up to 15 digits accurately after decimal point.

12) What is a Unicode system ?

Ans). Unicode system is an encoding standard that provides a unique number for every character, no matter what the platform, program, or language is. Unicode uses 2 bytes to represent a single character.

13) How are positive and negative numbers represented internally ?

Ans). Positive numbers are represented in binary using 1's complement notation and negative numbers are represented by using 2's complement notation.

14) What is the difference between >> and >>> ?

Ans). Both bitwise right shift operator(>>) and bitwise zero fill right shift operator(>>>) are used to shift the bits towards right. The difference is that >> will protect the sign bit whereas the >>> operator will not protect the sign bit. It always fills 0 in the sign bit.

15) What are control statements ?

Ans). Control statements are the statements which alter the flow of execution and provide better control to the programmer on the flow of execution. They are useful to write better and complex programs.

16) Out of do..while and while - - which loop is efficient ?

Ans). In a do..while loop, the statements are executed without testing the condition, the first time. From the second time only the condition is observed. This means that the programmer does not have control right from the beginning of its execution. In a while loop, the condition is tested first and then only the statements are executed. This means it provides better control right from the beginning. Hence, while loop is more efficient than do..while loop.

17) What is a collection ?

Ans). A collection represents a group of elements like integer values or objects. Examples for collections are arrays and java.util_classes (stack, LinkedList, Vector, etc).

18) Why goto statements are not available in Java ?

Ans). Goto statements lead to confusion for a programmer. Especially in a large program, if several goto statements are used, the programmer would be perplexed while understanding the flow from where to where the control is jumping.

19) What is the difference between return and System.exit(0) ?

Ans). Return statement is used inside a method to come out of it. System.exit(0) is used in any method to come out of the program.

20) What is the difference between System.out.exit(0) and System.exit(1) ?

Ans). System.exit(0) terminates the program normally. Whereas System.exit(1) terminates the program because of some error encountered in the program.

21) What is the difference between System.out, System.err and System.in?

Ans). System.out and System.err both represent the monitor by default and hence can be used to send data or results to the monitor. But System.out is used to display normal messages and results whereas System.err is used to display error messages and System.in represents InputStream object, which by default represents standard input device, ie, keyboard.

22) On which memory, arrays are created in Java?

Ans). Arrays are created on dynamic memory by JVM. There is no question of static memory in Java; every thing(variables, array, object etc.) is created on dynamic memory only.

23) Can you call the main() method of a class from another class ?

Ans). Yes , we can call the main() method of a class from another class using Classname.main() . At the time of calling the main() method, we should pass a string type array to it.

24) Is String a class or data type ?

Ans). String is a class in java.lang package. But in Java, all classes are also considered as data types. So we can take String as a data type also.

25) Can we call a class as a data type ?

Ans). Yes, a class is also called ‘ user-defined’ data type. This is because a use can create a class.

26) What is object reference ?

Ans). Object reference is a unique hexadecimal number representing the memory address of the object. It is useful to access the members of the object.

27) What is difference between == and equals() while comparing strings ? which one is reliable ?

Ans). == operator compares the references of the sting objects. It does not compare the contents of the objects. equals() method compares the contents. While comparing the strings, equals() method should be used as it yields the correct result.

28) What is a string constant pool ?

Ans). String constant pool is a separate block of memory where the string objects are held by JVM. If a sting object is created directly, using assignment operator as: String s1 = “Hello”,then it is stored in string constant pool.

29) Explain the difference between the following two statements:

1. String s=”Hello”

2. String s = new String(“Hello”);

Ans). In the first statement, assignment operator is used to assign the string literal to the String variables. In this case, JVM first of all checks whether the same object is already available in the string constant pool. If it is available , then it creates another reference to it. If the same object is not available , then it creates another object with the content “ Hello “ and stores it into the string constant pool. In the second statement, new operator is used to create the string object, in this case, JVM always creates a new object without looking in the string constant pool.

30) What is the difference between String and StringBuffer classes?

Ans). String class objects are immutable and hence their contents cannot be modified. StringBuffer class objects are mutable, so they can be modified. Moreover the methods that directly manipulate data of the object are not available in String class. Such methods are available in StringBuffer class.

31) Are there any other classes whose objects are immutalbe ?

Ans). Yes, classes like Character, Byte, Integer, Float, Double, Long..called ‘wrapper classes’ are created as ‘immutable’.Classes like Class, BigInteger, Big Decimal are also immutable.

32) What is the difference between StringBuffer and StringBuilder classes?

Ans). StringBuffer class is synchronized and StringBuilder is not. When the programmer wants to use several threads, he should use StringBuffer as it gives reliable results . If only one thread is used. StringBuilder is preferred, as it improves execution time.

33) What is object oriented approach ?

Ans). Object oriented programming approach is a programming methodology to design computer programs using classes and objects.

34) What is the difference between a class and an object ?

Ans). A class is a model for creating objects and does not exist physically. An object is any thing that exists physically. Both the classes and objects contain variables and methods.

35) What is encapsulation ?

Ans). Encapsulation is a mechanism where the data(variables) and the code(methods) that act on the data will bind together. For ex, if we take a class, we write the variables and methods inside the class. Thus, class is binding them together. So class is an example for encapsulation.

36) What is abstraction ?

Ans). Hiding the unnecessary data from the user and expose only needed data is of interest to the user. A good example for abstraction is a car. Any car will have some parts like engine, radiator, mechanical and electrical equipment etc. The user of the car (driver) should know how to drive the car and does not require any knowledge of these parts. For example driver is never bothered about how the engine is designed and the internal parts of the engine. This is why, the car manufacturers hide these parts from the driver in a separate panel, generally at the front.

Example in java:

Class Bank

```
{
Private int accno;
Private String name;
Private float balance;
Private float profit;
Private float loan;
Public void display_to _clerk()
{
System.out.println("Accno= "+accno);
System.out.println("Name=" +name);
System.out.println("Balance="+balance);
}
}
```

37) What is Inheritance ?

Ans). It creates new classes from existing classes, so that the new classes will acquire all the features of the existing classes is called inheritance. (or) Acquiring the all properties from base class to child class .

38) What is Polymorphism ?

Ans). The word 'Polymorphism' came from two Greek words 'poly' meaning 'many' and 'morphs' meaning 'forms' . Thus, polymorphism represents the ability to assume several different forms. In programming, we can use a single variable to refer to objects of different types and thus, using that variable we can call the methods of the different objects. Thus a method call can perform different tasks depending on the type of the object.

39) What is the difference between object oriented programming languages and object based programming languages ?

Ans). Object oriented programming languages follow all the features of Object Oriented Programming System(OOPS). Smalltalk, Simula-67, C++, Java are examples for OOPS languages. Object based programming languages follow all the features of OOPS except Inheritance. For example, JavaScript and VBScript will come under object based programming languages.

40) What is hash code?

Ans). Hash code is unique identification number allotted to the objects by the JVM. This hash code number is also called reference number which is created based on the location of the object in memory, and is unique for all objects, except for String objects.

41) How can you find the hash code of an object ?

Ans). The hashCode() method of 'Object' class in java.lang.package is useful to find the hash code of an object.

42) Can you declare a class as 'private' ?

Ans). No, if we declare a class as private , then it is not available to java compiler and hence a compile time error occurs, But inner classes can be declared as private.

43) When is a constructor called, before or after creating the object ?

Ans). A Constructor is called concurrently when the object creation is going on. JVM first allocates memory for the object and then executes the constructor to initialize the instance variables. By the time, object creation is completed, the constructor execution is also completed.

44) What is the difference between default constructor and parameterized constructor?

Default constructor Parameter constructor

Default constructor is useful to initialize all objects with same data. Parameterized constructor is useful to initialize each object with different data. Default constructor does not have any parameters. Parameterized constructor will have 1 or more parameters When data is not passed at the time of creating an object, default constructor is called. When data is passed at the time of creating an object parameterized constructor is called.

45) What is the difference between a constructor and a method?

Constructors Methods

A constructor is used to initialize the instance variables of a class. A method is used for any general purpose processing or calculations. A constructor's name and class name should be same. A method's name and class name can be same or different. A constructor is called at the time of creating object. A method can be called after creating the object. A constructor is called only once per object. A method can be called several times on the object.

46) What is constructor overloading ?

Ans). Writing two or more constructors with the same name but with difference in the parameters is called constructor overloading. Such constructors are useful to perform different tasks.

47) What are instance methods ?

Ans). Instance methods are the methods which act on the instance variables of the class. To call the instance methods , we should use the form; *objectname.methodname()*.

Ex: double x = obj.sum();

48) What are static methods ?

Ans). Static methods are the methods which do not act upon the instance variables of a class. Static methods are declared as 'static'.

49) What is the difference between instance variables and class variables(static variables) ?

Ans). 1. An Instance variable is a variable whose separate copy is available to each object. A class variable is a variable whose single copy in memory is shared by all objects.

2. Instance variables are created in the objects on heap memory. Class variables are stored on method area.

50) Why instance Variables are not available to static methods ?

Ans). After executing static methods, JVM creates the objects. So the instance variables of the objects are not available to static methods.

51) Is it possible to compile and run a Java program without writing main() method ?

Ans). Yes , it is possible by using a static block in the Java program.

52) How are objects are passed to methods in Java ?

Ans). Primitive data types, objects, even object references – every thing is passed to methods using ‘pass by value’ or ‘call by value’ concept. This means their bit by bit copy is passes to the methods.

53) What are factory methods ?

Ans). A factory methods is a method that creates and returns an object to the class to which it belongs. A single factory method replaces several constructors in the class by accepting different options from the user, while creating the object.

54) In how many ways can you create an object in Java ?

Ans). There are four ways of creating objects in Java:

1. Using *new* operator

Employee obj = new Employee();

Here , we are creating Employee class object ‘obj’ using new operator.

2. Using factory methods:

Number Format obj = NumberFormat. getNumberInstance();

Here, we are creating *NumberFormat* object using the factory method *getNumberInstance()*

3. Using *newInstance()* method. Here we should follow tow steps, as:

(a) First, store the class name ‘Employee’ as a string into an object. For this purpose, factory metod *forName()* of the class ‘Class’ will be useful:

Class c = Class.forName(“Employee”);

We should note that there is a class with the name ‘Class’ in *java.lang* package.

(b) Next, create another object to the class whose name is in the object c. For this purpose , we need *newInstance()* method of the class ‘Class’ as:

Employee obj = (Employee)c.newInstance();

4. By cloning an already available object, we can create another object. Creating exact copy of an existing object is called ‘cloning’.

Employee obj1 = new Employee ();

Employee obj2 = (Employee)obj1.clone();

Earlier, we created obj2 by cloning the Employee object obj1.*clone()* method of Object class is used to clone object.We should note that there is a class by the name ‘Object’ in *java.lang* package.

55) What is object graph ?

Ans). Object graph is a graph showing relationship between different objects in memory.

56) What is anonymous inner class ?

Ans). It is an inner class whose name is not written in the outer class and for which only one object is created.

57) What is Inheritance ?

Ans). Deriving new classes from existing classes such that the new classes acquire all the features of existing classes is called inheritance.

58) Why super class members are available to sub class ?

Ans). Because, the sub class object contains a copy of super class object.

59) What is the advantage of inheritance ?

Ans). In inheritance a programmer reuses the super class code without rewriting it, in creation of sub classes So, developing the classes becomes very easy. Hence, the programmer's productivity is increased.

60) Why multiple inheritance is not available in Java ?

Ans). Multiple inheritance is not available in Java for the following reasons:

1. It leads to confusion for a Java program.
2. The programmer can achieve multiple inheritance by using interfaces.
3. The programmer can achieve multiple inheritance by repeatedly using single inheritance.

61) How many types of inheritance are there?

Ans). There are two types of inheritances single and multiple. All other types are mere combinations of these two. However, Java supports only single inheritance.

62) What is coercion ?

Ans). Coercion is the automatic conversion between different data types done by the compiler.

63) What is conversion ?

Ans). Conversion is an explicit change in the data type specified by the operator.

64) What is method signature ?

Ans). Method signature represents the method name along with method parameters.

65) What is method overloading ?

Ans). Writing two or more methods in the same class in such a way that each method has same name but with different method signatures – is called method overloading.

66) What is method overriding ?

Ans). Writing two or more methods in super and sub classes such that the methods have same name and same signature is called method overriding.

67) What is the difference between method overloading and method overriding ?**MethodOverloading Method Overriding**

Writing two or more methods with the same name but with different signatures is called method overloading. Writing two or more methods with the same name and same signatures is called method overriding. Method overloading is done in the same class. Method overriding is done in super and sub classes. In method overloading, method return type can be same or different. In method overriding method return type should also be same. JVM decides which method is called depending on the difference in the method signatures. JVM decides which method is called depending on the data type (class) of the object used to call the method. Method overloading is done when the programmer wants to extend the already available features. Method overriding is done when the programmer wants to provide a different implementation(body) for the same feature. Method overloading is code refinement. Same method is refined to perform a different task. Method overriding is code replacement. The sub class method overrides(replaces) the super class method.

68) Can you override private methods ?

Ans). No, private methods are not available in the sub classes, so they cannot be overridden.

69) Can we take private methods and final methods as same ?

Ans). Yes. The Java compiler assigns the value for the private methods at the time of compilation. Also private methods can not be modified at run time. This is the same cases with final methods also. Neither the private methods nor the final methods can be overridden . So, private methods can be taken as final methods.

70) What is final?

Ans). 'final' keyword is used in two ways:

- It is used to declare constants as:

Final double PI = 3.14159; // PI is constant

It is used to prevent inheritance as:

- *Final class A // sub class to A cannot be created.*

71) What is the difference between dynamic polymorphism and static polymorphism?

Ans). Dynamic polymorphism is the polymorphism existed at runtime. Here, Java compiler does not understand which method is called at compilation time. Only JVM decides which method is called at runtime. Method overloading and method overriding using instance methods are the examples for dynamic polymorphism. Static polymorphism is the polymorphism exhibited at compile time. Here, Java compiler knows which method is called. Method overloading and method overriding using static methods; method overriding using private or final methods are examples for static polymorphism.

72) What is difference between primitive data types and advanced data types?

Ans). Primitive data types represent single values. Advanced data types represent a group of values. Also methods are not available to handle the primitive data types. In case of advanced data types, methods are available to perform various operations.

73) What is implicit casting?

Ans). Automatic casting done by the Java compiler internally is called implicit casting. Implicit casting is done to convert a lower data type into a higher data type.

74) What is explicit casting?

Ans). The casting done by the programmer is called explicit casting. Explicit casting is compulsory while converting from a higher data type to a lower data type.

75) What is generalization and specialization?

Ans). Generalization is a phenomenon where a sub class is promoted to a super class, and hence becomes more general. Generalization needs widening or up-casting. Specialization is a phenomenon where a super class is narrowed down to a sub class. Specialization needs narrowing or down-casting.

76) What is widening and narrowing?

Ans). Converting lower data type into a higher data type is called widening and converting a higher data type into a lower type is called narrowing. Widening is safe and hence even if the programmer does not use cast operator, the Java compiler does not flag any error. Narrowing is unsafe and hence the programmer should explicitly use cast operator in narrowing.

77) Which method is used in cloning?

Ans). *clone()* method of Object class is used in cloning.

78) What do you call the interface without any members?

Ans). An interface without any members is called marking interface or tagging interface. It marks the class objects for a special purpose. For example, *Cloneable*(java.lang) and *Serializable*(java.io) are two marking interfaces. *Cloneable* interface indicates that a particular class objects are cloneable while *Serializable* interface indicates that a particular class objects are serializable.

79) What is abstract method?

Ans). An abstract method is a method without method body. An abstract method is written when the same method has to perform difference tasks depending on the object calling it.

80) What is abstract class?

Ans). An abstract class is a class that contains 0 or more abstract methods.

81) How can you force your programmers to implement only the features of your class?

Ans). By writing an abstract class or an interface.

82) Can you declare a class as *abstract* and *final* also?

Ans). No, *abstract* class needs sub classes. *final* key word represents sub classes which cannot be created. So, both are quite contradictory and cannot be used for the same class.

83) What is an interface?

Ans). An interface is a specification of method prototypes, All the methods of the interface are public and abstract.

84) Why the methods of interface are public and abstract by default?

Ans). Interface methods are public since they should be available to third party vendors to provide implementation. They are abstract because their implementation is left for third party vendors.

85) Can you implement one interface from another?

Ans). No, we can't implement an interface means writing body for the methods. This cannot be done again in an interface, since none of the methods of the interface can have body.

86) Can you write a class within an interface?

Ans). Yes, it is possible to write a class within an interface.

87) Explain about interfaces?

Ans). * An interface is a specification of method prototypes, before we proceed further, written in the interface without method bodies.

*An interface will have 0 or more abstract methods which are all public and abstract by default.

* An interface can have variables which are public static and final by default. This means all the variables of the interface are constants.

88) What is the difference between an abstract class and an interface?

Abstract class Interface

An abstract class is written when there are some common features shared by all the objects. An interface is written when all the features are implemented differently in different objects. When an abstract class is written, it is the duty of the programmer to provide sub classes to it.

An interface is written when the programmer wants to leave the implementation to the third party vendors. An abstract class contains some abstract methods and also some concrete methods. An interface contains only abstract methods. An abstract class contain instance variables also. An interface cannot contain instance variables. It contains only constants. All the abstract methods of the abstract class should be implemented in its sub classes. All the (abstract) methods of the interface should be implemented in its implementation classes. Abstract class is declared by using the keyword *abstract*. Interface is declared using the keyword *interface*.

89) A programmer is writing the following statements in a program:

1. *import java.awt.*;*
2. *import java.awt.event.*;*

Should he write both the statements in his program or the first own is enough?

Ans). *event* is a sub package of *java.awt* package. But, when a package is imported, its sub packages are not automatically imported into a program. So, for every package or sub package, a separate import statement should be written. Hence if the programmer wants the classes and interfaces of both the *java.awt* and *java.awt.event* packages, then he should both the preceding statements in his program.

90) How can you call the garbage collector?

Ans). We can call garbage collector of JVM to delete any unused variables and unreferenced objects from memory using *gc()* method. This *gc()* method appears in both *Runtime* and *System* classes of *java.lang* package. For example, we can call it as:

```
System.gc();  
Runtime.getRuntime().gc();
```

91) What is the difference between the following two statements.

1. *import pack.Addition;*
2. *import pack.*;*

Ans). In statement 1, only the *Addition* class of the package *pack* is imported into the program and in statement 2, all the classes and interfaces of the package *pack* are available to the program.

If a programmer wants to import only one class of a package say *BufferedReader* of *java.io* package, we can write *import java.io.BufferedReader;*

92) What is the differentiate between .ear, .jar and .war files.?

Ans). These files are simply zipped file using *java jar* tool. These files are created for different purposes. Here is the description of these files: .jar files: These files are with the .jar extension. The .jar files contains the libraries, resources and accessories files like property files. .war files: These files are with the .war extension. The war file contains the web application that can be deployed on the any servlet/jsp container. The .war file contains jsp, html, javascript and other files for necessary for the development of web applications. .ear files: The .ear file contains the EJB modules of the application.

93) What is CLASSPATH?

Ans). The *CLASSPATH* is an environment variable that tells the Java compiler where to look for class files to import. *CLASSPATH* is generally set to a directory or a JAR(Java Archive)file.

94) What is a JAR file?

Ans) A Java Archive file (JAR) is a file that contains compressed version of several .class files, audio files, image files or directories. JAR file is useful to bundle up several files related to a project and use them easily.

95) What is the scope of default access specifier ?

Ans). Default members are available within the same package, but not outside of the package. So their scope is package scope.

96) What happens if `main()` method is written without `String args[]` ?

Ans). The code compiles but JVM cannot run it, as it cannot see the `main()` method with `String args[]`.

97). What are checked exceptions ?

Ans). The exceptions that are checked at compilation-time by the Java compiler are called 'checked exceptions'. The exceptions that are checked by the JVM are called 'unchecked exceptions'.

98). What is Throwable ?

Ans). Throwable is a class that represents all errors and exceptions which may occur in Java.

99). Which is the super class for all exceptions ?

Ans). Exception is the super class of all exceptions in Java.

100). What is the difference between an exception and an error ?

Ans). An exception is an error which can be handled. It means when an exception happens, the programmer can do something to avoid any harm. But an error is an error which cannot be handled, it happens and the programmer cannot do any thing.

101). What is the difference between `throws` and `throw` ?

Ans). *throws* clause is used when the programmer does not want to handle the exception and throw it out of a method. *throw* clause is used when the programmer wants to throw an exception explicitly and wants to handle it using catch block. Hence, *throws* and *throw* are contradictory.

102). Is it possible to re-throw exceptions ?

Ans). Yes, we can re-throw an exception from catch block to another class where it can be handled.

103). Why do we need wrapper classes ?

1. They convert primitive data types into objects and this is needed on Internet to communicate between two applications.

2. The classes in *java.util* package handle only objects and hence wrapper classes help in this case also.

104). Which of the wrapper classes contains only one constructor ? (or) Which of the wrapper classes does not contain a constructor with `String` as parameter ?

Ans). Character.

105). What is unboxing ?

Ans). Converting an object into its corresponding primitive datatype is called unboxing.

106). What happens if a string like "Hello" is passed to `parseInt()` method ?

Ans). Ideally a string with an integer value should be passed to *parseInt()* method. So, on parsing "Hello", an exception called "NumberFormatException" occurs since the *parseInt()* method cannot convert the given string "Hello" into an integer value.

107). What is a collection framework ?

Ans). A collection framework is a class library to handle groups of objects. Collection framework is implemented in java.util.package.

108). Does a collection object store copies of other objects or their references ?

Ans). A Collection object stores references of other objects.

109). Can you store a primitive data type into a collection ?

Ans). No, Collections store only objects.

110). What is the difference between Iterator and ListIterator ?

Ans). Both are useful to retrieve elements from a collection. Iterator can retrieve the elements only in forward direction. But Listener can retrieve the elements in forward and backward direction also. So ListIterator is preferred to Iterator.

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

Ans). Both are useful to retrieve elements from a collection. Iterator has methods whose names are easy to follow and Enumeration methods are difficult to remember. Also Iterator has an option to remove elements from the collection which is not available in Enumeration. So, Iterator is preferred to Enumeration.

112). What is the difference between a Stack and LinkedList ?

Ans). 1. A Stack is generally used for the purpose of evaluation of expression. A LinkedList is used to store and retrieve data.

2. Insertion and deletion of elements only from the top of the Stack is possible. Insertion and deletion of elements from any where is possible in case of a LinkedList.

113). What is the difference between ArrayList and Vector ?

ArrayList Vector

ArrayList object is not synchronized by Vector object is synchronized by default. default

Incase of a single thread, using ArrayList is faster than the Vector.

In case of multiple threads, using Vector is advisable. With a single thread, Vector becomes slow. ArrayList increases its size every time by 50 percent (half). Vector increases its size every time by doubling it.

114). Can you synchronize the ArrayList object ?

Ans). Yes, we can use synchronizedList() method to synchronize the ArrayList, as:
Collections.synchronizedList(new ArrayList());

115). What is the load factor for a HashMap or Hashtable ?

Ans). 0.75.

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

Ans).

HashMap Hashtable

HashMap object is not synchronized by default. Hashtable object is synchronized by default.

In case of a single thread, using HashMap is faster than the Hashtable.

In case of multiple threads, using Hashtable is advisable, with a single thread, Hashtable becomes slow.

HashMap allows null keys and null values to be stored.

Hashtable does not allow null keys or values.

Iterator in the HashMap is fail-fast. This means

Iterator will produce exception if concurrent updates are made to the HashMap.

Enumeration for the Hashtable is not fail-fast. This means even if concurrent updations are done to Hashtable, there will not be any incorrect results produced by the Enumeration.

117). Can you make HashMap synchronized ?

Ans). Yes, we can make HashMap object synchronized using synchronizedMap() method as shown here:

```
Collections.synchronizedMap(new HashMap( ));
```

118). What is the difference between a Set and a List ?

Ans).

Set List

A set represents a collection of elements. Order of the elements may change in the set.

A List represents ordered collection of elements. List preserves the order of elements in which they are entered.

Set will not allow duplicate values to be stored. List will allow duplicate values.

Accessing elements by their index (position number) is not possible in case of sets.

Accessing elements by index is possible in lists.

Sets will not allow null elements. Lists allow null elements to be stored.

119). What is the difference between System.out and System.err ?

Ans). Both are used to display messages on the monitor. System.out is used to display normal messages

As:

```
System.out.println("This is nayanimalidhar");
```

```
System.err.println("This is an error");
```

120). What is the advantage of stream concept..?

Ans). Streams are mainly useful to move data from one place to another place. This concept can be used to receive data from an input device and send data to an output device.

121). What is the default buffer size used by any buffered class ?

Ans). 512 bytes.

122). What is serialization ?

Ans). Serialization is the process of storing object contents into a file. The class whose objects are stored in the file should implement 'serializable' interface of java.io.package.

123).What type of variables cannot be serialized ?

Ans). Static and transient variables cannot be serialized.

Once the objects are stored into a file, they can be later retrieved and used as and when needed. This is called de-serialization.

124). What is IP address ?

Ans). An IP address is a unique identification number allocated to every computer on a network or Internet. IP address contains some bytes which identify the network and the actual computer inside the network.

125). What is DNS ?

Ans). Domain Naming Service is a service on Internet that maps the IP address with corresponding website names.

126). What is a socket ?

Ans). A socket is a point of connection between a server and a client on a network.

127). What is port number ?

Ans). Port number is a 2 byte number which is used to identify a socket uniquely.

128). Which thread always runs in a Java program by default ?

Ans). main thread. A thread represents execution of statements. The way the statements are executed is of two types: 1). Single tasking 2). Multi tasking.

129). Why threads are called light-weight ?

Ans). Threads are light-weight because they utilize minimum resources of the system. This means they take less memory and less processor time.

130). What is the difference between single tasking and multitasking ?

Ans). Executing only one job at a time is called single tasking. Executing several jobs at a time is called multi tasking. In single tasking, the processor time is wasted, but in multi tasking, we can utilize the processor time in an optimum way.

131). How can you stop a thread in Java ?

Ans). First of all , we should create a boolean type variable which stores ' false' . When the user wants to stop the thread. We should store 'true' into the variable. The status of the variable is checked in the run () method and if it is true, the thread executes 'return' statement and then stops.

132). What is the difference between 'extends Thread' and 'implements Runnable' ? Which one is advantageous ?

Ans). extends Thread and implements Runnable – both are functionally same. But when we write extends Thread, there is no scope to extend another class, as multiple inheritance is not supported in Java. Class MyClass extends Thread, AnotherClass //invalid If we write implements Runnable, then still there is scope to extend another class. class MyClass extends AnotherClass implements Runnable //valid

This is definitely advantageous when the programmer wants to use threads and also wants to access the features of another class.

133). Which method is executed by the thread by default ?

Ans). public void run() method.

134). What is Thread synchronization ?

Ans). When a thread is already acting on an object, preventing any other thread from acting on the same object is called 'Thread synchronization' or 'Thread safe' The object on which the threads are synchronized is called 'synchronized object'. Thread synchronization is recommended when multiple threads are used on the same object(in multithreading).

135). What is the difference between synchronized block and synchronized keyword ?

Ans). Synchronized block is useful to synchronize a block of statements. Synchronized keyword is useful to synchronize an entire method.

138). What is Thread deadlock ?

Ans). When a thread has locked an object and waiting for another object to be released by another thread, and the other thread is also waiting for the first thread to release the first object, both the threads will continue waiting forever. This is called 'Thread deadlock'.

139). What is the difference between the sleep() and wait() methods ?

Ans). Both the sleep() and wait() methods are used to suspend a thread execution for a specified time. When sleep() is executed inside a synchronized block, the object is still under lock. When wait() method is executed, it breaks the synchronized block, so that the object lock is removed and it is available. Generally, sleep() is used for making a thread to wait for some time. But wait() is used in connection with notify() or notifyAll() methods in thread communication.

140). What is the default priority of a thread ?

Ans). When a thread is created, by default its priority will be 5.

141). What is demon thread ?

Ans). A daemon thread is a thread that executes continuously. Daemon threads are service providers for other threads or objects. It generally provides a background processing.

142). What is thread life cycle ?

Ans). A thread is created using new Thread() statement and is executed by start() method. The thread enters 'runnable' state and when sleep() or wait() methods are used or when the thread is blocked on I/O, it then goes into 'not runnable' state. From 'not runnable' state, the thread comes back to the 'runnable' state and continues running the statements. The thread dies when it comes out of run() method. These state transitions of a thread are called 'life cycle of a thread'.

143). What is the difference between a window and a frame ?

Ans). A window is a frame without any borders and title, whereas a frame contains borders and title.

144). What is event delegation model ?

Ans). Event delegation model represents that when an event is generated by the user on a component, it is delegated to a listener interface and the listener calls a method in response to the event. Finally, the event is handled by the method.

145). Which model is used to provide actions to AWT components ?

Ans). Event delegation model.

146). What is an adapter class ?

Ans). An adapter class is an implementation class of a listener which contains all methods implemented with empty body. For example, WindowAdapter is an adapter class of WindowListener interface. Adapter classes reduce overhead on programming while working with listener interfaces.

147). What is anonymous inner class ?

Ans). Anonymous inner class is an inner class whose name is not mentioned, and for which only one object is created.

148). What is the default layout in a frame ?

Ans). BorderLayout.

149). What is the default layout in an applet ?

Ans). FlowLayout.

150). What are Java Foundation classes ?

Ans). Java Foundation classes (JFC) represented a class library developed in pure Java which is an extension to AWT.

151). Discuss about the MVC architecture in JFC/ swing ?

Ans). Model- View – Controller is a model used in swing components. Model represents the data of the component. View represents its appearance and controller is a mediator between the model and the view. MVC represents the separation of model of an object from its view and how it is controlled.

152). What are the various window panes available in swing ?

Ans). There are 4 window panes: Glass pane, Root pane, Layered pane, and Content pane.

153). Where are the borders available in swing ?

Ans). All borders are available in BorderFactory class in javax.swing.border package.

154). What is an applet ?

Ans). An applet represents Java byte code embedded in a web page.

155). What is applet life cycle ?

Ans). An applet is born with `init()` method and starts functioning with `start()` method. To stop the applet, the `stop()` method is called and to terminate the applet completely from memory, the `destroy()` method is called. Once the applet is terminated, we should reload the HTML page again to get the applet start once again from `init()` method. This cyclic way of executing the methods is called applet life cycle.

156). Where are the applets executed ?

Ans). Applets are executed by a program called applet engine which is similar to virtual machine that exists inside the web browser at client side.

157). What is HotJava?

Ans). Hot Java is the first applet-enabled browser developed in Java to support running of applets.

158). Which tag is used to embed an applet into a HTML page ?

Ans). `<APPLET>` tag is used to insert an applet into HTML page.

159). What is a generic type ?

Ans). A generic type represents a class or an interface that is type-safe. It can act on any data type.

160). What is erasure ?

Ans). Creating non-generic version of a generic type by the Java compiler is called erasure.

161). What is auto boxing ?

Ans). Auto boxing refers to creating objects and storing primitive data types automatically by the compiler.

162). What is JDBC ?

Ans). JDBC (Java Database Connectivity) is an API that is useful to write Java programs to connect to any database, retrieve the data from the database and utilize the data in a Java program.

163). What is a database driver ?

Ans). A database driver is a set of classes and interfaces, written according to JDBC API to communicate with a database.

164). How can you register a driver ?

Ans). To register a database driver, we can follow one of the 4 options:

- By creating an object to driver class
- By sending driver class object to `DriverManager.registerDriver()` method
- By sending the driver class name to `Class.forName()` method
- By using `System` class `getProperty()` method.