1) What is Java?

A) Java is an object oriented programming language

2) Is Java invented for Internet?

A) No, it was developed for programming towards tiny devices

3) What is byte code?

A) It is machine understandable code and targeted for JVM (Java Virtual Machine). It is class file.

4) What is JVM?

A) Java Virtual Machine which accepts java byte code and produces result

5) What are java buzzwords?

A) Java buzzwords explain the important features of java. They are Simple, Secured, Portable, architecture neutral, high performance, dynamic, robust, interpreted etc.

6) Is byte code is similar to .obj file in C?

A) Yes, both are machine understandable codes No, .obj file directly understood by machine, byte code requires JVM

7) What is new feature in control statements comparing with C/C++?

A) Labeled break and labeled continue are new

8) What are new features in basic features comparing with C/C++?

A) Data types: All data types on all machines have fixed size;

Constants: final is used to declare constant

Boolean Type: boolean is new data type in Java which can store

true/falseThere are no structures/unions/pointers

9) Is String data type?

A) No, Strings are classes in Java (Arrays are classes)

10) What are length and length( ) in Java?

A) Both gives number of char/elements, length is variable defined in Arrayclass, length( ) is method defined in String class Object-Orientation Section

11) What is class?

A) Class is blue print for objects; Class is collection of objects; Class gives the general structure for objects.

12) What is object?

A) Object is an instance of class. Object is real world entity which has state, identity and behavior.

13) What is encapsulation?

A) Encapsulation is packing of data with their methods

14) What is abstraction?

A) Abstraction is the process of identification of essential features of objects

15) What is data hiding?

A) Implementation details of methods are hidden from the user

16) What is hierarchy?

A) Ordering of classes (or inheritance)

17) What is inheritance?

A) Extending an existing class is called inheritance. Extracting the features of super class

18) Why inheritance is important?

A) Reusability is achieved through inheritance

19) Which types of inheritances are available in Java?

A) Simple, Hierarchical and Multilevel

20) Can we achieve multiple inheritances in Java?

A) With the help of interfaces we can achieve

21) What is interface?

A) Interface is collection of final variables and abstract methods

(We need not give final and abstract keywords and By default they are public methods)

22) What is the difference between class and interface?

A) Class contains implemented methods, but interface contains non implemented methods

23) What is polymorphism?

A) Multiple (poly) forms (morphs) same instance can respond different manners is called polymorphism

24) Can we achieve run time polymorphism in Java?

A) Yes, also called as dynamic polymorphism. Possible by dynamic method dispatch

25) What is dynamic method dispatch?

A) When you assign object of sub class for the super class instance, similar methods of super class are hidden

26) What is overloading?

A) Same method name can be used with different type and number of arguments (in same class)

27) What is overriding?

A) Same method name, similar arguments and same number of arguments can be defined in super class and sub class. Sub class methods override the super class methods.

28) What is the difference between overloading and overriding?

A) Overloading related to same class and overriding related sub-super class Compile time polymorphism achieved through overloading, run time polymorphism achieved through overriding

Keywords Section

29) What is keyword?

A) Java reserved word which should is not used as variable/class-name (e.g.:break, for, if,while etc)

30) What is final keyword?

A) Used before variables for declaring constants Used before methods for preventing from overriding .Used before class-name for preventing from inheritance

31) What is static keyword?

A) Used before variable for defining shared variables.Used before method for defining class-level methods, these methods Can be called without creating objects (e.g.: parseInt method of Integer class)

32) What is abstract keyword?

A) Used for creating abstract class, class which doesn't have any instance

33) What is this keyword?

A) To call current class variables and current class methods we can use this

key word

34) What is this( )?

A) Used for calling another constructor of current class

35) What is super keyword?

A) To call super class variables and super class methods we can use super key word

36) What is super( )?

A) Used for calling of super class constructor and it should be first executable

statement insub class constructor

Packages and Exception handling section

37) What is package?

A) Package is collection of classes and interfaces

38) How to define package?

A) By using package keyword before the definition of class

39) What is CLASSPATH?

A) It is an environment variable which is used for defining the location of class

files

40) What is jar?

A) Jar stands for Java archive files, compressed set of class files and can beused in CLASSPATH

41) What is the meaning of import java.awt.*;?

A) Import all the classes and interfaces in the java.awt.package. This doesn't imports other packages defined in java.awt. package.

42) Is it necessary to give import java.awt.event.*;, when already import java.awt.* given?

A) Yes, import java.awt.* doesn't imports event package defined in java.awt.package.

43) What is exception?

A) Abnormal termination of program is called exception

44) What is exception handler?

A) A method which controls abnormal termination of program

45) What are the keywords used for exception handling?

A) try, catch, throw, throws and finally

46) What is the difference between throw and throws keywords?

A) throw keyword is used for invoking an exception (For raising) throws keyword lists exception names which can be ignored from the method execution

47) What is the difference between final and finally?

A) final is used for defining constants

finally is used for executing code after completion of exception handler

48) What is catch( ) block?

A) Exception handler that controls abnormal termination of program

49) Can we have multiple catch( ) blocks?

A) Yes, but first sub class exceptions should be handled

50) What happen is we not handle sub class exception first?

A) Generates compile time error saying that unreachable code is defined in

the program

Multithreading Section

51) What is thread?

A) Thread is a part of process, also called as light weight process

52) What is multithreading?

A) Simultaneous execution of threads is called multithreading

53) What is a process?

A) Program under execution is called process

54) How to create thread?

A) By creating instance of Thread class or implementing Runnable interface

55) Can you name any methods of Thread class?

A) currentThread( ), setName( ), getName( ), setPriority( ), getPriority( ), join( ), isAlive( )

56) What is join( ) method of Thread?

A) Combines two or more threads running process and wait till their execution is completed

57) What are the states in Thread Life Cycle?

A) Ready, running, block, wait, dead Applets, AWT Section

58) What is an applet?

A) Applet is a java program which runs via java-enabled browser

59) Is graphics possible only in applets?

A) No, stand-alone program frames can also display graphics

60) What is the relation between java and Internet?

A) With the help of java applets, we can write programming for Internet

61) Which package is required to write GUI (Graphical User Interface) programs?

A) Java.awt

62) Which package is required to write Applets?

A) Java.applet

63) What is an event?

A) A kind of action

64) What is event handling?

A) A procedure which gives functionality for the events

65) How to implement event handling?

A) By using interfaces like ActionListener, MouseListener

66) What is the method available in ActionListener Interface?

A) public void action performed (AcitonEvent e)

67) How to pass parameters to Applets?

A) By using PARAM tag.

68) Types of Inheritance in OOPS

A) Different types of inheritances in OOPs are as follows:

1. Single Inheritance

2. Multiple Inheritance

3. Multi-level Inheritance

4. Multi-path Inheritance

5. Hierarchical Inheritance

6. Hybrid Inheritance

69) What are the limitations of OOPs?

A) Following are some of the common limitations of OOPs:

Size exceeds that of other programs.

It took a lot of work to make, and it runs more slowly than other programs.

It is inappropriate for certain types of issues.

It takes some getting used to.

70) What are constructors?

A) The constructor has the same name as the class.

A constructor is also a special kind of method. It is used to initialize objects of the class.

71) Types of constructor

A) Types of constructors depend upon languages

Private Constructor

Default Constructor

Copy Constructor

Static Constructor

Parameterized Constructor

72) Explain different ways of creating a thread. Which one would you prefer and why?

A) There are three ways that can be used in order for a Thread to be created:

• A class may extend the Thread class.

• A class may implement the Runnable interface.

• An application can use the Executor framework, in order to create a thread pool.

The Runnable interface is preferred, as it does not require an object to inherit the Thread class. In case your application design requires multiple inheritance, only interfaces can help you. Also, the thread pool is very efficient and can be implemented and used very easily.

73) Explain the available thread states in a high-level.

A) During its execution, a thread can reside in one of the following states:

• Runnable: A thread becomes ready to run, but does not necessarily start

running immediately.

• Running: The processor is actively executing the thread code.

• Waiting: A thread is in a blocked state waiting for some external processing

to finish.

• Sleeping: The thread is forced to sleep.

• Blocked on I/O: Waiting for an I/O operation to complete.

• Blocked on Synchronization: Waiting to acquire a lock.

• Dead: The thread has finished its execution.

74) What is the difference between a synchronized method and a synchronized block?

A) In Java programming, each object has a lock. A thread can acquire the lock for an object by using the synchronized keyword.

The synchronized keyword can be applied in a method level (coarse grained lock) or block level of code (fine grained lock).

75) How does thread synchronization occurs inside a monitor? What levels of synchronization can you apply?

A) The JVM uses locks in conjunction with monitors. A monitor is basically a guardian that watches over a sequence of synchronized code and ensuring that only one thread at a time executes a synchronized piece of code. Each monitor is associated with an object reference. The thread is not allowed to execute the code until it obtains the lock.

Collections section

What is Collection in Java?

In **Java**, a collection is a framework that provides an architecture for storing and manipulating a collection of objects. In JDK 1.2, a new framework called "Collection Framework" was created, which contains all of the collection classes and interfaces.

Collections in Java are capable of doing any data operations such as searching, sorting, insertion, manipulation, and deletion.

A single unit of objects in Java is referred to as a collection. The two basic "root" interfaces of Java collection classes are the Collection interface (java.util.Collection) and the Map interface(java.util.Map). Many interfaces (Set, List, Queue, Deque) and classes are available in the Java Collection **framework** (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

Differentiate between Collection and collections in the context of Java.

Collection : In the java.util.package, there is an interface called a collection. It's used to represent a collection of separate objects as a single entity. It's equivalent to the container in the C++ programming language. The collection framework's root interface is referred to as the collection. It has a number of classes and interfaces for representing a collection of individual objects as a single unit. The key sub-interfaces of the collection interface are List, Set, and Queue. Although the map interface is part of the Java collection framework, it does not inherit the interface's collection. The Collection interface's most significant functions are add(), remove(), clear(), size(), and contains().

Collections: The java.util.package has a utility class called Collections. It defines various utility methods for working with collections, such as sorting and searching. All of the methods are static. These techniques give developers much-needed convenience, allowing them to interact with Collection Framework more successfully. It provides methods like sort() to sort the collection elements in the normal sorting order, and min() and max() to get the minimum and maximum value in the collection elements, respectively.

ArrayList	**LinkedList**
The elements of this class are stored in a dynamic array. This class now supports the storage of all types of objects thanks to the addition of generics.	The elements of this class are stored in a doubly-linked list. This class, like the ArrayList, allows for the storage of any type of object.
The List interface is implemented by this class. As a result, this serves as a list.	The List and Deque interfaces are both implemented by this class. As a result, it can be used as both a list and a deque.
Because of the internal implementation, manipulating an ArrayList takes longer. Internally, the array is scanned and the memory bits are shifted whenever we remove an element.	Because there is no concept of changing memory bits in a doubly-linked list, manipulating it takes less time than manipulating an ArrayList. The reference link is changed after traversing the list.
This class is more useful when the application requires data storage and access.	This class is more useful when the application requires data manipulation.

What is Class IN JAVA

"Class" (uppercase):

- This is a built-in class in Java, part of the java.lang package.
- It represents the class itself at runtime.
- You can use it to get information about a class, such as its name, superclass, methods, fields, etc.