DURGA's INTERVIEW QUESTIONS

1.Can a lock be acquired on a class?

Yes, a lock can be acquired on a class. This lock is acquired on the class's Class object. The synchronized keyword tells the JVM that the method requires a lock in order to run. The JVM then creates the lock and manages the allocation of the lock to threads during execution. To execute static synchronized area compulsory the thread has to get the class level lock.

2.Can main method be declared final?

Yes, the main method can be declared final, in addition to being public static Inaddintion to final we can declared main method with synchronized modifier also The following is valid main method declaration

3. Can a public class MyClass be defined in a source file named YourClass.java?

If java source file contains a public class then the name of the source file and name of the public class must be matched . otherwise compilation fails.

4.What is the default value of the local variables?

The local variables are not initialized to any default value, neither primitives nor object references. If you try to use these variables without initializing them explicitly, the java compiler will not compile the code. It will complain about the local variable not being initialized. Hence,
                    Before using local variable it must be initialized

5. Is Empty.java file a valid source file?

  Yes, an empty .java file is a perfectly valid source file.

6. What are the different scopes for Java variables?

The scope of a Java variable is determined by the context in which the variable is declared. Thus a java variable can have one of the three scopes at any given point in time.

1. Instance : - These are typical object level variables, they are initialized to default values at the time of creation of object, and remain accessible as long as the object accessible.

2. Local : - These are the variables that are defined within a method. They remain accessbile only during the course of method execcution. When the method finishes execution, these variables fall out of scope.

3. Static: - These are the class level variables. They are initialized when the class is loaded in JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular object instance.

7. What happens if you dont initialize an instance variable of any of the primitive types in Java?

Java by default initializes it to the default value for that primitive type. For the integral types the default value is 0, for floating point data types default value is 0.0 and for the booelan types it is false.

8.What will be the initial value of an object reference, which is defined as an instance variable?

The object references are all initialized to null in Java. However in order to do anything useful with these references, you must set them to a valid object, else you will get NullPointerExceptions everywhere you try to use such default initialized references.

9. Are main, next, delete and exit keywords in Java?

No, they are not keywords in Java. delete is not a keyword in Java. Java does not make use of explicit destructors the way C++ does. To exit a program explicitly you use exit method in System object.

10. Is String a primitive data type in Java?

No String is not a primitive data type in Java, even though it is one of the most extensively used object. Strings in Java are instances of String class defined in java.lang package.

11.What does it mean that a method or field is "static"?

static variables are class level variables. Only one copy will be created per class and shared by all the objects. We can access static variables either by class name or by object reference but recommended way is using class name. If you change the value of a static variable by using any object reference , it will automatically reflect for all the object references. That's how library methods like System.out.println() work out is a static field in the java.lang.System class Static methods are class level methods. Mostly these are utility methods we can access without creating objects.Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too).

12.Give a few reasons for using Java?

Built-in support for multi-threading, socket communication, and memory management (automatic garbage collection).Object Oriented (OO). Better portability than other languages across operating systems. Supports Web based applications (Applet, Servlet, and JSP), distributed applications (sockets, RMI, EJB etc) and network protocols (HTTP, JRMP etc) with the help of extensive standardized APIs (Application Programming Interfaces).

13.What are "static initializers" or "static blocks with no function names"?

When a class is loaded, all blocks that are declared static and don't have function name (i.e. static initializers) are executed even before the constructors are executed. As the name suggests they are typically used to initialize static fields.

14. How do you express an 'is a' relationship and a 'has a' relationship or explain inheritance and composition? What is the difference between composition and aggregation?

The 'is a' relationship is expressed with inheritance and 'has a' relationship is expressed with composition. Both inheritance and composition allow you to place sub-objects inside your new

class. Two of the main techniques for code reuse are class inheritance and object composition. Inheritance is uni-directional. For example House is a Building. But Building is not a House. Inheritance uses extends key word. Composition: is used when House has a Bathroom. It is incorrect to say House is a Bathroom. Composition simply means using instance variables that refer to other objects. The class House will have an instance variable, which refers to a Bathroom object

15. What is the main difference between a String and a StringBuffer class?
String is immutable: you can't modify a string object but can replace it by creating a new instance. Creating a new instance is rather expensive.
StringBuffer is mutable: use StringBuffer or StringBuilder when you want to modify the contents. StringBuilder was added in Java 5 and it is identical in all respects to StringBuffer except that it is not synchronized, which makes it slightly faster at the cost of not being thread-safe.

16. What is the difference between processes and threads?
A process is an execution of a program but a thread is a single execution sequence within the process.
A process can contain multiple threads. A thread is sometimes called a lightweight process.

17. Explain different ways of creating a thread?
Threads can be used by either :
Extending the Thread class
Implementing the Runnable interface.

18.What are transient variables in java?
Transient variables are variable that cannot be serialized.

19.What is synchronization?

Synchronization is the ability to control the access multiple threads to shared resources.
Synchronization stops multithreading. With synchronization , at a time one thread will be able to access a shared resource.

20.What is the difference between yield() and sleep()?

When a object invokes yield() it returns to ready state for giving the chance to remaining threads of same priority. But when an object invokes sleep() method enters into sleeping state. After speeping time expires then only thread entered into ready state

21. What is finalize() method?

Garbage Collector calls this method just before destroying any object to perform clean up activities.

22.Can we call finalize() method ?

Yes. Nobody will stop us to call any method , if it is accessible in our class. But a garbage collector cannot call an object's finalize method if that object is reachable.

23.Can we declare an anonymous class as both extending a class and implementing an interface?
No. An anonymous class can extend a class or implement an interface but not both simultaneously.

24.What is the use of "bin" and "lib" in JDK?

"bin" contains all tools such as javac, appletviewer, awt tool, etc. whereas "lib" contains API and all packages

25.What are the legal operands of the instanceof operator?

The left operand is an object reference or null value and the right operand is a class, interface, or array type.

26.How is it possible for two String objects with identical values not to be equal under the == operator?

The == operator compares two objects to determine if they are the same object in memory. It is possible for two String objects to have the same value, but located indifferent areas of memory.

27. What is final, finalize() and finally?

final : final keyword can be used for class, method and variables. A final class cannot be subclassed and it prevents other programmers from subclassing a secure class to invoke insecure methods. A final method can't be overridden. A final variable can't change from its initialized value.
finalize() : finalize() method is used just before an object is destroyed and can be called just prior to garbage collection.
finally : finally, a key word used in exception handling, creates a block of code that will be executed after a try/catch block has completed and before the code following the try/catch block. The finally block will execute whether or not an exception is thrown. For example, if a method opens a file upon exit, then you will not want the code that closes the file to be bypassed by the exception-handling mechanism. This finally keyword is designed to address this contingency.

28.What is the relationship between a method's throws clause and the exceptions that can be thrown during the method's execution?
A method's throws clause must declare any checked exceptions that are not caught within the body of the method. These exceptions must be handled by the caller.

29. How many ways can an argument be passed to a subroutine?

An argument can be passed in two ways. They are Pass by Value and Passing by Reference.
Passing by value: This method copies the value of an argument into the formal parameter of the subroutine.
Passing by reference: In this method, a reference to an argument (not the value of the argument) is passed to the
parameter.

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

31.When can an object reference be cast to an interface reference?

An object reference be cast to an interface reference when the object implements the referenced interface.

32.How many methods in the Externalizable interface?

There are two methods in the Externalizable interface. You have to implement these two methods in order to make your class externalizable. These two methods are
readExternal() and writeExternal()

33. What are synchronized methods and synchronized statements?

Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

34. Can an unreachable object become reachable again?
An unreachable object may become reachable again. This can happen when the object's finalize() method is invoked and the object performs an operation which causes it to become accessible to reachable objects.

35. How do I serialize an object to a file?

The class whose instances are to be serialized should implement an interface Serializable. Then you pass the instance to the ObjectOutputStream which is connected to a Fileoutputstream. This will save the object to a
file.

36. Which methods of Serializable interface should I implement?

The serializable interface is an empty interface, it does not contain any methods. So we do not implement any methods

37. How can I customize the seralization process? i.e. how can one have a control over the serialization process?

Yes it is possible to have control over serialization process. The class should implement Externalizable interface. This interface contains two methods namely read External and writeExternal. You should implement these methods and write the logic for customizing the serialization process.

38. What is the common usage of serialization?

Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serilazed.

39. When you serialize an object, what happens to the object references included in the object?

The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process. Thus when an object is serialized, all the included objects are also serialized alongwith the original obect.

40. What one should take care of while serializing the object?

One should make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.

41. What happens to the static fields of a class during serialization?

Serialization ignores static fields, because they are not part of any particular state.

42.Does importing a package imports the subpackages as well?

No you will have to import the subpackages explicitly.
For eg: Importing com.MyTest.* will import classes in the package MyTest only. It will not import any class in any of it's subpackage.

43.If I write return at the end of the try block, will the finally block still execute?

Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return.

44.Is it necessary that each try block must be followed by a catch block?

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

45.What are the different ways to handle exceptions?

There are two ways to handle exceptions,
1. **By using try-catach** : By wrapping the desired code in a try block followed by a catch block to

catch the exceptions. and
2. **By using throws clause:** List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions

46.What are runtime exceptions?

Runtime exceptions are those exceptions that are thrown at runtime because of either wrong input data or because of wrong business logic etc. These are not checked by the compiler at compile time.

47.What are checked exceptions?
Checked exception are those which the Java compiler forces you to catch for smooth execution of program at runtime. e.g. IOException are checked Exceptions.

48.Does Java provide any construct to find out the size of an object?
No there is not sizeof operator in Java. So there is not direct way to determine the size of an object directly in Java.

49.What is Externalizable interface?
Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism. Thus if your class implements this interface, you can customize the serialization process by implementing these methods

50.What is the purpose of garbage collection in Java, and when is it used?

The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources can be reclaimed and reused. A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

51.What is the difference between a constructor and a method?

A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.

52.What if the main method is declared as private?

The program compiles properly but at runtime it will give "Main method not public." message.

53.What is an Iterator interface?

The Iterator interface is used to step through the elements of a Collection.

54.Describe the Garbage Collection process in Java ?

The JVM spec mandates automatic garbage collection outside of the programmers control. The

System.gc() or Runtime.getRuntime().gc() is merely a suggestion to the JVM to run the GC process but is NOT guaranteed.

55.What is constructor chaining and how is it achieved in Java ?

A child object constructor always first needs to construct its parent (which in turn calls its parent constructor.). In Java it is done via an implicit call to the no-args constructor as the first statement

56.What methods can be overridden in Java?

All Java methods can be overwritten in subclasses except those that are declared final, static, and private.

57.What is daemon thread and which method is used to create the daemon thread?
Daemon thread is a low priority thread which runs intermittently in the back ground for providing the support to Non-Daemon . setDaemon method is used to create a daemon thread.

58.What are Transient and Volatile Modifiers?
Transient: The transient modifier applies to variables only and it is not stored as part of its object's Persistent state. Transient variables are not serialized.
Volatile: Volatile modifier applies to variables only and it tells the compiler that the variable modified by volatile can be changed unexpectedly by other parts of the program. JVM maintains a separate copy of the volatile variable for every thread.

59.What is the difference between instanceof and isInstance?

instanceof is used to check to see if an object can be cast into a specified type without throwing a classcastexception. isInstance() determines if the specified Object is assignment-compatible with the object represented by this Class. This method is the dynamic equivalent of the Java language instanceof operator. The method returns true if the specified Object argument is non-null and can be cast to the reference type represented by this Class object without raising a ClassCastException. It returns false otherwise.

60.What does the "abstract" keyword mean in front of a method? A class?

Abstract keyword declares either a method or a class. If a method has a abstract keyword in front of it,it is called abstract method.Abstract method has no body.It has only arguments and return type.Abstract methods act as placeholder methods that are implemented in the subclasses.
Abstract classes can't be instantiated.If a class is declared as abstract,no objects of that class can be created.If a class contains any abstract method it must be declared as abstract

61.What is JDBC? Describe the steps needed to execute a SQL query using JDBC.

The JDBC is a pure Java API used to execute SQL statements. It provides a set of classes and interfaces that can be used by developers to write database applications.
The steps needed to execute a SQL query using JDBC:

1. Open a connection to the database.
2. Execute a SQL statement.
3. Process th results.
4. Close the connection to the database.

62.What are native methods? How do you use them?
Native methods are methods whose implementation is provided in another programming language such as C. The main objective of native methods are to improve the performance of the system.

63.How can you force all derived classes to implement a method present in the base class?

Creating and implementing an interface would be the best way for this situation. Just create an interface with empty methods which forces a programmer to implement all the methods present under it. Another way of achieving this task is to declare a class as abstract with all its methods abstract.

64.What's the difference between == and equals method?

The equals method can be considered to perform a deep comparison of the value of an object, whereas the == operator performs a shallow comparison.
The equals() method compares the characters inside a string object. == operator compares two object references to check whether they refer to the same instances or not.

65. How can you achieve Multiple Inheritance in Java?
Java's interface mechanism can be used to implement multiple inheritance, with one important difference from c++ way of doing MI: the inherited interfaces must be abstract. This obviates the need to choose between different implementations, as with interfaces there are no implementations.

66.What are the two basic ways in which classes that can be run as threads may be defined?

A thread class may be declared as a subclass of Thread, or it may implement the Runnable interface.

67.How are this() and super() used with constructors?

this() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

68.What restrictions are placed on the values of each case of a switch statement?

 the switch argument is any type of variable is allowed which can be implicitly promoted to int type. The case labels must be compile time constants.

69.What are order of precedence and associativity, and how are they used?
Order of precedence determines the order in which operators are evaluated in expressions. Associatity determines whether an expression is evaluated left-to-right or right-to-left

70.What is a StringTokenizer ?
String Tokenizer provide parsing process in which it identifies the delimiters provided by the user , by default delimiters are spaces, tab, newline etc. and separates them from the tokens. Tokens are those which are separated by
delimiters.

71.What are nested classes ?

There are two types : static and non-static.
static class means the members in its enclosing class (class within class) can be accessed by creating an object and cannot be accessed directly without creating the object.
non-static class means inner class and can be accessed directly with the object created for the outer class no need to create again an object like static class.

72.What is the difference between abstract class and interface?

a) All the methods declared inside an interface are abstract whereas abstract class may or may not contain abstract methods.
b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods.
c)Abstract class must have subclasses whereas interface can't have subclasses

73.Under what conditions is an object's finalize() method invoked by the garbage collector?

The garbage collector invokes an object's finalize() method just before destroying any method.

74.What are the steps involved for making a connection with a database or how do you connect to a database?

a) **Loading the driver** : To load the driver, Class. forName() method is used. Class. forName("sun. jdbc. odbc. JdbcOdbcDriver"); When the driver is loaded, it registers itself with the java. sql. DriverManager class as an available database driver.
b) **Making a connection with database**: To open a connection to a given database, DriverManager. getConnection() method is used. Connection con = DriverManager. getConnection ("jdbc:odbc:somedb", "user", "password");
c) **Executing SQL statements** : To execute a SQL query, java. sql. statements class is used. createStatement() method of Connection to obtain a new Statement object. Statement stmt = con. createStatement(); A query that returns data can be executed using the executeQuery() method of Statement. This method executes the statement and returns a java. sql. ResultSet that encapsulates the retrieved data: ResultSet rs = stmt. executeQuery("SELECT * FROM some table");
d) **Process the results** : ResultSet returns one row at a time. Next() method of ResultSet object can be called to move to the next row. The getString() and getObject() methods are used for retrieving column values: while(rs. next()) { String event = rs. getString("event"); Object count = (Integer) rs. getObject("count").

75.What is serialization and deserialization?

Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

76.What is the difference between JDBC and ODBC?

a) OBDC is for Microsoft and JDBC is for Java applications.
b) ODBC can't be directly used with Java because it uses a C interface.
c) ODBC makes use of pointers which have been removed totally from Java.
d) ODBC mixes simple and advanced features together and has complex options for simple queries.
But JDBC is designed to keep things simple while allowing advanced capabilities when required.
e) ODBC requires manual installation of the ODBC driver manager and driver on all client machines.
JDBC drivers are written in Java and JDBC code is automatically installable, secure, and portable on all platforms.
f) JDBC API is a natural Java interface and is built on ODBC. JDBC retains some of the basic features of ODBC.

77.What is the difference between Reader/Writer and InputStream/Output Stream?

The Reader/Writer class is character-oriented and the InputStream/OutputStream class is byte-oriented.

78.What is a stream and what are the types of Streams and classes of the Streams?
A Stream is an abstraction that either produces or consumes information. There are two types of Streams and they are:
Byte Streams: Provide a convenient means for handling input and output of bytes.
Character Streams: Provide a convenient means for handling input & output of characters.
Byte Streams classes: Are defined by using two abstract classes, namely *InputStream* and *OutputStream*.
Character Streams classes: Are defined by using two abstract classes, namely *Reader* and *Writer*.

79.What are Vector, Hashtable, LinkedList and Enumeration?

Vector : The Vector class provides the capability to implement a growable array of objects.
Hashtable : The Hashtable class implements a Hashtable data structure. A Hashtable indexes and stores objects in a dictionary using hash codes as the object's keys. Hash codes are integer values that identify objects.
LinkedList: Removing or inserting elements in the middle of an array can be done using LinkedList. A LinkedList stores each object in a separate link whereas an array stores object references in consecutive locations.
Enumeration: An object that implements the Enumeration interface generates a series of elements, one at a time. It has two methods, namely hasMoreElements() and nextElement(). HasMoreElemnts() tests if this enumeration has more elements and nextElement method returns successive elements of the series

80.What is source and listener?
Source : A source is an object that generates an event. This occurs when the internal state of that object changes in some way.
Listener : A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.

81.What is the lifecycle of an applet?

init() method - Can be called when an applet is first loaded
start() method - Can be called each time an applet is started.
paint() method - Can be called when the applet is minimized or maximized.
stop() method - Can be used when the browser moves off the applet's page.
destroy() method - Can be called when the browser is finished with the applet.

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

83.What is the immediate superclass of the Applet class?
Panel

84.Can an object's finalize() method be invoked while it is reachable?

An object's finalize() method cannot be invoked by the garbage collector while the object is still reachable. However, an object's finalize() method may be invoked by other objects.

85.What is the difference between the >> and >>> operators?

The >> operator carries the sign bit when shifting right. The >>> zero-fills bits that have been shifted out.

86.What modifiers may be used with an inner class that is a member of an outer class?

A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

87.Which characters may be used as the second character of an identifier, but not as the first character of an identifier?

The digits 0 through 9 may not be used as the first character of an identifier but they may be used after the first character of an identifier

88.What is synchronization and why is it important?

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.

89.What is connection pooling?

With servlets, opening a database connection is a major bottleneck because we are creating and tearing down a new connection for every page request and the time taken to create connection will be more. Creating a connection pool is an ideal approach for a complicated servlet. With a connection pool, we can duplicate only the resources we need to duplicate rather than the entire servlet. A connection pool can also intelligently manage the size of the pool and make sure each connection remains valid. A number of connection pool packages are currently available. Some like DbConnectionBroker are freely available from Java Exchange Works by creating an object that dispenses connections and connection Ids on request. The ConnectionPool class maintains a Hastable, using Connection objects as keys and Boolean values as stored values. The Boolean value indicates whether a connection is in use or not. A program calls getConnection() method of the ConnectionPool for getting Connection object it can use; it calls returnConnection() to give the connection back to the pool.

90.What are the types of statements in JDBC?

Statement: to be used createStatement() method for executing single SQL statement
PreparedStatement — To be used preparedStatement() method for executing same SQL statement over and over.
CallableStatement — To be used prepareCall() method for multiple SQL statements over and over.

91.What is daemon thread and which method is used to create the daemon thread?

Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.

92. What is the class and interface in java to create thread and which is the most advantageous method?

Thread class and Runnable interface can be used to create threads and using Runnable interface is the most advantageous method to create threads because we need not extend thread class here.

93. What are the methods for inter-thread communication and what is the class in which these methods are defined?

wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class.
wait() : When a thread executes a call to wait() method, it surrenders the object lock and enters into a waiting state.
notify() or notifyAll() : To remove a thread from the waiting state, some other thread must make a call to notify() or notifyAll() method on the same object.

94.What is multithreading?

Multithreading is the mechanism in which more than one thread run independent of each other within the process.

95. What is the difference between process and thread?

Process is a program in execution whereas thread is a separate path of execution in a program.
What is the difference between Array and vector?
Array is a set of related data type and static whereas vector is a growable array of objects and dynamic.

96. What is the difference between String and String Buffer?

String objects are constants and immutable whereas StringBuffer objects are not.
b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.

97. What are inner class and anonymous class?
Inner class : classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private. Anonymous class : **Anonymous class** is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors.

98 . Can you have an inner class inside a method and what variables can you access?

Yes, we can have an inner class inside a method and final variables can be accessed.

99. What is a cloneable interface and how many methods does it contain?

It is not having any method because it is a TAGGED or MARKER interface.

100. How many times may an object's finalize() method be invoked by the garbage collector?

An object's finalize() method may only be invoked once by the garbage collector.

101. What modifiers may be used with top-level class?
public, abstract and final,strictfp can be used for top-level class.

102. What is the difference between the Boolean & operator and the && operator?
If an expression involving the Boolean & operator is evaluated, both operands are evaluated. Then the & operator is applied to the operand. When an expression involving the && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then the second operand is evaluated. The && operator is then applied to the first and second operands. If the first operand evaluates to false, the evaluation of the second operand is skipped.