1. How garbage collector knows that the object is not in use and needs to be removed?

Ans- Garbage collector reclaimed objects that are no longer being used, clears their memory, and keeps the memory available for future allocation.

Any unreferenced object is a garbage and will be collected.

# 2)What is the purpose of garbage collection?

The garbage collection process is to identify the objects which are no longer referenced or needed by a program so that their resources can be reclaimed and reused. These identified objects will be discarded.

|  |
| --- |
| **3)Question: Give the list of Java Object class methods.** |
|  |
| **Answer:** | | |
| clone() - Creates and returns a copy of this object.  euals() - Indicates whether some other object is "equal to" this one.  finalize() - Called by the garbage collector on an object when garbage collection  determines that there are no more references to the object.  getClass() - Returns the runtime class of an object.  hashCode() - Returns a hash code value for the object.  notify() - Wakes up a single thread that is waiting on this object's monitor.  notifyAll() - Wakes up all threads that are waiting on this object's monitor.  toString() - Returns a string representation of the object.  wait() - Causes current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object. | | |

# 4)Question: Can we override static method?

|  |
| --- |
| **Answer:** |
| We cannot override static methods. Static methods are belogs to class, not belongs  to object. Inheritance will not be applicable for class members |

# 5)Question: Can you list serialization methods?

Serialization interface does not have any methods. It is a marker interface.

It just tells that your class can be serializable.

# 6)Question: What is the difference between super() and this()?

super() is used to call super class constructor, whereas this() used to call

constructors in the same class, means to call parameterized constructors.

# How to prevent a method from being overridden?

By specifying final keyword to the method you can avoid overriding in a subcalss. Similarlly one can use final at class level to prevent creating subclasses.

# Can we create abstract classes without any abstract methods?

Yes, we can create abstract classes without any abstract methods.

# Can we have static methods in interface?

By default, all methods in an interface are decleared as public, abstract. It will never be static. But this concept is changed with java 8. Java 8 came with new feature called "default methods" with in interfaces .

Before Java 8, the interface only contains method signatures. With Java 8 new feature Default Methods or Defender Methods, you can include method body within the interface.

**package com.java2novice.dm;**

**public interface FirstInterface {**

**default void someMethod(){**

**System.out.println("from FirstInterface...");**

**}**

**}**

# What is transient variable?

Transient variables cannot be serialized. During serialization process, transient variable states will not be serialized. State of the value will be always defaulted after deserialization.

# Incase, there is a return at the end of try block, will execute finally block?

Yes, the finally block will be executed even after writing return statement at the end of try block. It returns after executing finally block.

# What is abstract class or abstract method?

We cannot create instance for an abstract class. We can able to create

instance for its subclass only. By using abstract keyword just before

class, we can make a class as abstract class.

public abstract class MyAbstractClass{

}

Abstract class may or may not contains abstract methods. Abstract method is

just method signature, it does not containes any implementation. Its subclass

must provide implementation for abstract methods. Abstract methods are looks

like as given below:

public abstract int getLength();

# What is default value of a boolean?

Default value of a boolean is false.

# When to use LinkedList or ArrayList?

Accessing elements are faster with ArrayList, because it is index based.But accessing is difficult with LinkedList. It is slow access. This isto access any element, you need to navigate through the elements one by one.

But insertion and deletion is much faster with LinkedList, because if you know the node, just change the pointers before or after nodes. Insertion and deletion is slow with ArrayList, this is because, during these operations ArrayList need to adjust the indexes according to deletion or insetion if you are performing on middle indexes. Means, an ArrayList having 10 elements, if you are inserting at index 5, then

you need to shift the indexes above 5 to one more.

# What is daemon thread?

Daemon thread is a low priority thread. It runs intermittently[নিরবচ্ছিন্ন ] in the back ground, and takes care of the garbage collection operation for the java runtime system. By calling setDaemon() method is used to create a daemon thread.

# Does each thread in java uses seperate stack?

In Java every thread maintains its own separate stack. It is called Runtime Stack but they share the same memory.

# What is the difference between Enumeration and Iterator?

The functionality of Enumeration and the Iterator are same.

You can get remove() from Iterator to remove an element, while Enumeration does not have remove()method.

Using Enumeration you can only traverse and fetch the objects, where as using Iterator we can also add and remove the objects. So Iterator can be useful if you want to manipulate the list and Enumeration is for read-only access.

# Find out below switch statement output.

|  |
| --- |
| **Code:** |
| public static void main(String a[]){  int price = 6;  switch (price) {  case 2: System.out.println("It is: 2");  default: System.out.println("It is: default");  case 5: System.out.println("It is: 5");  case 9: System.out.println("It is: 9");  }  } |
| Answer: |
| It is: default  It is: 5  It is: 9 |

# Does system.exit() in try block executes code in finally block?

|  |
| --- |
| **Code:** |
| try{  System.out.println("I am in try block");  System.exit(1);  } catch(Exception ex){  ex.printStackTrace();  } finally {  System.out.println("I am in finally block!!!");  } |
| **Answer:** | |
| It will not execute finally block. The program will be terminated  after System.exit() statement. | |

# What is fail-fast in java?

A fail-fast system is immediately report any failure that is likely to lead to failure. When a problem occurs, a fail-fast system fails immediately.

In Java, we can find this behavior with iterators. Incase, you have called iterator on a collection object, and another thread tries to modify the collection object, then concurrent modification exception will be thrown. This is called fail-fast.

# What is final, finally and finalize?

final:

final is a keyword. The variable decleared as final should be

initialized only once and cannot be changed. Java classes

declared as final cannot be extended. Methods declared as final

cannot be overridden.

finally:

**Finally is a block.** The finally block always executes when the try block exits. This ensures that the finally block is executed even if an unexpected exception occurs. But finally is useful for more than just exception handling - it allows the programmer to avoid having cleanup code accidentally bypassed by a return, continue, or break. Putting cleanup code in a finally block is always a good practice, even when no exceptions are anticipated.

finalize:

finalize is a method. Before an object is garbage collected, the runtime system calls its finalize() method. You can write system resources release code in finalize() method before getting garbage collected.

|  |
| --- |
| **In java, are true and false keywords?**  true, false, and null might seem like keywords, but they are actually literals. You cannot use them as identifiers in your programs. |
|  |

# What are the types of ResultSet?

|  |  |
| --- | --- |
| |  | | --- | | The type of a ResultSet object determines the level of its functionality in  two areas: the ways in which the cursor can be manipulated, and how concurrent  changes made to the underlying data source are reflected by the ResultSet object.  The sensitivity of a ResultSet object is determined by one of three different  ResultSet types:  TYPE\_FORWARD\_ONLY:  The result set cannot be scrolled; its cursor moves forward only, from  before the first row to after the last row. The rows contained in the  result set depend on how the underlying database generates the results.  That is, it contains the rows that satisfy the query at either the time  the query is executed or as the rows are retrieved.    TYPE\_SCROLL\_INSENSITIVE:  The result can be scrolled; its cursor can move both forward and backward  relative to the current position, and it can move to an absolute position.  The result set is insensitive to changes made to the underlying data source  while it is open. It contains the rows that satisfy the query at either the  time the query is executed or as the rows are retrieved.    TYPE\_SCROLL\_SENSITIVE:  The result can be scrolled; its cursor can move both forward and backward  relative to the current position, and it can move to an absolute position.  The result set reflects changes made to the underlying data source while  the result set remains open.  The default ResultSet type is TYPE\_FORWARD\_ONLY. | |
| |  | | --- | |  | |

|  |
| --- |
| What is difference between wait and sleep methods in java? |

sleep():

It is a static method on Thread class. It makes the current thread into the

"Not Runnable" state for specified amount of time. During this time, the thread

keeps the lock (monitors) it has acquired.

wait():

It is a method on Object class. It makes the current thread into the "Not Runnable"

state. Wait is called on a object, not a thread. Before calling wait() method, the

object should be synchronized, means the object should be inside synchronized block.

The call to wait() releases the acquired lock.

# What happens if one of the members in a class does not implement Serializable interface?

When you try to serialize an object which implements Serializable interface, incase if the object includes a reference of an non serializable object then NotSerializableException will be thrown.

# How to get current time in milli seconds?

System.currentTimeMillis() returns the current time in milliseconds. It is a static method, returns long type.

# How can you convert Map to List?

We know that Map contains key-value pairs, whereas a list contains only objects. Since Entry class contains both key-value pair, Entry class will helps us to convert from Map (HashMap) to List (ArrayList). By using Map.entrySet() you will get Set object, which intern you can use it to convert to list object.

|  |
| --- |
| **Code:** |
| public static void main(String a[]){  Map<String, String> wordMap = new HashMap<String, String>();  Set<Entry<String, String>> set = wordMap.entrySet();  List<Entry<String, String>> list = new ArrayList<Entry<String, String>>(set);  } |

# What is strictfp keyword?

By using strictfp keyword, we can ensure that floating point operations take place precisely.

# What is System.out in Java?

Here out is an instance of PrintStream. It is a static member variable in System class. This is called standard output stream, connected to console.

# When to use String and StringBuffer?

String is immutable object. We can not change the value of a String object once it is initiated. If we try to change the value of the existing String object then it creates new object rather than changing the value of the existing object. So incase, we are going to do more modificatios on String, then use StringBuffer. StringBuffer updates the existing objects value, rather creating new object. StringBuffer is mutable object in java.

# What is difference between StringBuffer and StringBuilder?

The only difference between StringBuffer and StringBuilder is StringBuffer is thread-safe, that is StringBuffer is synchronized.

# Question: What is wrapper class in java?

|  |
| --- |
| **Answer:** |
| Everything in java is an object, except primitives. Primitives are int, short, long, boolean, etc. Since they are not objects, they cannot return as objects, and collection of objects. To support this, java provides  wrapper classes to move primitives to objects. Some of the wrapper classes are Integer, Long, Boolean, etc. |

# Is Iterator a Class?

Iterator is an interface. It is not a class. It is used to iterate through each and every element in a list. Iterator is implemented Iterator design pattern.

# What is java classpath?

The classpath is an environment variable. It is used to let the compiler know where the class files are available for import.

# Can a class in java be private?

We can not declare top level class as private. Java allows only public and default modifier for top level classes in java. Inner classes can be private.

Is null a keyword in java? The null value is not a keyword in java. true and false are also not keywords in java. They are reserved words in java language.

# What is the initial state of a thread when it is started?

When the thread is created and started, initially it will be in the ready state.

# What is the super class for Exception and Error?

The super class or base class for Exception and Error is Throwable.