**Garbage Collection:**

1. Which part of the memory is involved in Garbage Collection? Stack or Heap?

Heap.

1. What is responsibility of Garbage Collector?

Garbage collection is the process of automatically clearing the memory used by the objects that will be no longer be used by the program.

1. Garbage Collector is controlled by whom?

Invocation of GC is automatically determined by the JVM.

1. When does an object become eligible for garbage collection?

An object becomes eligible for Garbage Collection when no live thread can access it.

1. What is the purpose of finalize() method?

This method is used for garbage collection. This method is used for garbage collection. This method will get execute before Garbage Collection is done, because it release the resource which are held by the program.

1. If an object becomes eligible for Garbage Collection and its finalize() method has been called and inside this method the object becomes accessible by a live thread of execution and is not garbage collected. Later at some point the same object becomes eligible for Garbage collection, will the finalize() method be called again?

No

1. How many times does the garbage collector calls the finalize() method for an object?

Only once.

1. What happens if an uncaught exception is thrown from during the execution of the finalize() method of an object?

The exception will be ignored and the garbage collection (finalization) of that object terminates.

1. What are different ways to call garbage collector?

Garbage collection can be invoked using **System.gc() or Runtime.getRuntime().gc().**

1. How to enable/disable call of finalize() method of exit of the application?

**Runtime.getRuntime().runFinalizersOnExit(boolean value)** . Passing the boolean value will either disable or enable the finalize() call.

1. What are the different ways to make an object eligible for Garbage Collection when it is no longer needed?

Three different ways to make an object available for GC.

* Set all available object references to null.
* Make the reference variable to refer to another object
* Creating Islands of Isolation.

**Set all available object references to null:** once the purpose of creating the object is served:

|  |
| --- |
| public class GarbageCollnTest1 {  public static void main (String [] args){  String str = "Set the object ref to null";  //String object referenced by variable str is not eligible for GC yet  str = null;  /\*String object referenced by variable str becomes eligible for GC \*/  }  } |

**Make the reference variable to refer to another object:**

|  |
| --- |
| publc class GarbageCollnTest2 {  public static void main(String [] args){  String str1 = "Garbage collected after use";  String str2 = "Another String";  System.out.println(str1);  //String object referred by str1 is not eligible for GC yet  str1 = str2;  /\* Now the str1 variable referes to the String object "Another String" and the object "Garbage collected after use" is not referred by any variable and hence is eligible for GC \*/  }  } |

**Creating Islands of Isolation.**

|  |
| --- |
| public class GCTest3 {  GCTest3 g;  public static void main(String [] str){  GCTest3 gc1 = new GCTest3();  GCTest3 gc2 = new GCTest3();  gc1.g = gc2; //gc1 refers to gc2  gc2.g = gc1; //gc2 refers to gc1  gc1 = null;  gc2 = null;  //gc1 and gc2 refer to each other and have no other valid //references  //gc1 and gc2 form Island of Isolation  //gc1 and gc2 are eligible for Garbage collection here  }  } |