Garbage Collection
www.techfaq360.com

• State the behavior that is guaranteed by the garbage collection system, and write code that explicitly makes objects eligible for collection.

You can answer all the questions if you keep the following rules in mind.
• Only thing guaranteed by the GC mechanism is : IF an object is really being destroyed, it's finalize() method would already have been called.
Note: It doesn't say anything about when will an object be GCed etc.
• An object is eligible for garbage collection, if the only references to that object are from other objects that are also eligible for garbage collection.
Note: It doesn't say anything about circular references. It depends on the actual JVM implementation.
• You CANNOT precisely say when the GC thread will run. Neither can you make the GC thread to run when you want.
Note: You can call System.gc() etc. but this only requests the JVM to run the GC thread.
Some Points to Remember:
• You may set all the reference variables pointing to an object to null. This will enable the GC to collect this object. But that does not mean the object will really be GCed. It is possible that the GC thread may not run at all for the whole life of the program.

• finalize() Method: Signature : protected void finalize() throws Throwable { }
It is used to release system resources like File handles, Network connections etc. But not memory. Memory can only be release by the GC thread.

• All objects have a finalize method as it is implemented in the Object class. But unlike constructors, finalize() does not call super class's finalize(). So, it is advisable (NOT REQUIRED) to put super.finalize() in the code of your finalize() method so as to give a chance to the super class to cleanup it's resources.

• The order in which finalize methods are called may not reflect the order in which objects are destroyed.

• It will be called ONLY ONCE for an object by the garbage collector. If any exception is thrown in finalize, the object is still eligible for garbage collection (depends on the GC mechanism).

• You can resurrect an object by creating an active reference to it in this method.

• You may call finalize() explicitly, but it would be just like another method call and will not release the memory.

• finalize can be overloaded, but only the method with above mentioned signature will be called by the GC.