**Assignment: (Core Java)**

**7 : Constructors and Destructors**

**Que.1 Constructor Types (Default, Parameterized)**

**Ans.1**  A constructor is a special method used to initialize objects. types of constructor as under:

**Default Constructor**

* No parameters
* Can be provided by programmer or automatically by Java (if no constructor is defined)

**Parameterized Constructor**

* Takes arguments to initialize object with specific values

**Que.2 Copy Constructor (Emulated in Java)**

**Ans.2** In Java, there is no built-in copy constructor like in C++, but we can emulate it by creating a constructor that takes an object of the same class and copies its values.

**Que.3 Constructor Overloading**

**Ans.3** Constructor overloading in Java is a feature that allows a class to have multiple constructors, each with the same name (the class name) but different parameter lists. This enables the creation of objects in various ways, providing flexibility in object initialization.

Key aspects of constructor overloading:

* Multiple Constructors:
* Same Name:
* Different Parameter Lists:

**Que.4 Object Life Cycle and Garbage Collection**

**Ans.4**

**1. Object Life Cycle**

The life cycle of an object in Java refers to the stages it goes through from creation to eventual memory reclamation.

Stages of Object Life Cycle

1. **Creation**
   * An object is instantiated using the new keyword.
   * Memory is allocated on the heap.
2. **In Use**
   * The object is actively referenced and used by the program.
3. **Unreachable**
   * No live references point to the object.
   * Happens if:
     + A reference goes out of scope
     + A reference is explicitly set to null
     + Part of a cyclic reference chain without external references
4. **Eligible for Garbage Collection**
   * Once unreachable, the object becomes a candidate for garbage collection.
5. **Finalization (Optional & Deprecated)**
   * The finalize() method may be called before memory reclamation.
   * Deprecated in modern Java due to unpredictability and performance overhead.
6. **Reclamation**
   * The Garbage Collector (GC) reclaims memory, making it available for new objects.

2**. Garbage Collection (GC) in Java**

Garbage collection is automatic memory management in Java that frees memory occupied by objects no longer in use, reducing memory leaks and improving stability.

**How Garbage Collection Works**

Java uses algorithms such as Mark-and-Sweep with generational optimizations.

1. Mark Phase
   * GC identifies all "live" objects reachable from GC roots (local variables, static variables, active threads).
2. Sweep Phase
   * Unmarked objects are considered garbage. Memory occupied by these objects is reclaimed.
3. Compact Phase (Optional)
   * Some GCs move live objects together to avoid memory fragmentation.