Exercises on Memory and Object Lifecycle in Java

Exercise 1: Object Creation and Constructors (Basics)

a) Define what happens in memory when an object is created using the `new` keyword in Java.
b) Write a Java class `Student` with:

- Instance variables: `name`, `regNo`
- A constructor to initialize them
  c) Create two objects of the class and explain whether they share memory.

Exercise 2: Default vs Parameterized Constructors

Consider the following class:

```
class Book {
    String title;
    int pages;
}
```

a) What values will `title` and `pages` have when an object is created?
b) Modify the class to include a parameterized constructor.
c) Explain how constructors affect object initialization in memory.

Exercise 3: Java References

Study the code below:

```
class Test {
    int x;
}

public class Main {
    public static void main(String[] args) {
        Test t1 = new Test();
        t1.x = 10;

        Test t2 = t1;
        t2.x = 20;

        System.out.println(t1.x);
    }
}
```

a) Predict the output.
b) Explain the role of references in this program.
c) Draw a simple memory diagram showing heap and stack usage.

Exercise 4: Object Eligibility for Garbage Collection

Consider the code:

```
class Demo {
    int value;
}

public class GCExample {
    public static void main(String[] args) {
        Demo d1 = new Demo();
        Demo d2 = new Demo();

        d1 = null;
        d2 = d1;
    }
}
```

a) At what point does an object become eligible for garbage collection?
b) How many objects are eligible for garbage collection at the end of the program?
c) Justify your answer.

Exercise 5: Garbage Collection Concepts

a) Define garbage collection in Java.
b) Explain **two conditions** under which an object becomes eligible for garbage collection.
c) Why does Java not allow programmers to explicitly destroy objects?

Exercise 6: `finalize()` Method (Conceptual)

a) What is the purpose of the `finalize()` method?
b) Why is the use of `finalize()` discouraged in modern Java?
c) State one alternative mechanism for resource management.

Exercise 7: Constructors and Object Lifecycle Order

Given the following code:

```
class Sample {
    Sample() {
        System.out.println("Constructor called");
    }
}

public class TestLifeCycle {
    public static void main(String[] args) {
        Sample s = new Sample();
        s = null;
        System.gc();
    }
}
```

a) Identify the stages of the object lifecycle illustrated in this program.
b) Is garbage collection guaranteed to occur after calling `System.gc()`? Explain.

Exercise 8: Pass-by-Value and Object References

Consider:

```
class Number {
    int value;
}

public class PassTest {
    static void modify(Number n) {
        n.value = 50;
    }

    public static void main(String[] args) {
        Number obj = new Number();
        obj.value = 10;
        modify(obj);
        System.out.println(obj.value);
    }
}
```

a) Predict the output.
b) Explain how Java handles object references when passed to methods.
c) Does Java support pass-by-reference? Justify your answer.

Exercise 9: Stack vs Heap Memory

a) Distinguish between stack memory and heap memory in Java.
b) State **two items** stored in stack memory and **two items** stored in heap memory.
c) Why are local variables destroyed before objects?

Exercise 10: Short Answer

Answer **any five**:

1. What is an object lifecycle in Java?
2. What is a constructor?
3. What happens if no constructor is defined in a class?
4. What is a reference variable?
5. State one role of the garbage collector.
6. What does `System.gc()` do?

NB: Use Java How to program as your reference book