Assignment No - 27

Q1] What is mean by object class?

→ The object class in java is the superclass of all classes. It is defined in the java.lang package, and every class in java implicitly inherits from it, either directly or indirectly.

- Since java follows an inheritance hierarchy, every class you create is subclass of object, unless you explicitly extend another class.

• Features of object class.

1] Superclass of All classes.

- Every java class is child of object.
- Even if a class does not explicitly extend another class, it still extends object.

2] Provides Common Methods

- The object class defines several methods that all Java objects inherit.
- These methods include toString(), equals(), hashCode()

3] Supports polymorphism

- You can store any object reference in an object type variable.

Q2] Explain mark and sweep algorithm of garbage collection

→ The Mark and Sweep algorithm is one of the fundamental garbage collection techniques used in programming languages like java to reclaim memory occupied by objects that are no longer reachable.

• The algorithm works in two main phase.

1] Mark Phase

- The garbage collector starts from root objects
- It traverses the object graph, making all objects

that are reachable from these roots.
- Objects that cannot be reached remain unmarked
2) Sweep Phase
- The GC scans through the heap memory and removes all unmarked objects.
- The freed memory is then reused for new objects

Q3] What are the types of methods in which we can use this keyword?
→ The this keyword in java is a reference variable that refers to the current object. It can be used in different types of methods to distinguish between instance variables, invoke constructors and call other methods.
• Where this can used?
1) Referring to Instance Variables - Used when local and instance variables have the same name.
2) Calling methods of the same class - Used to explicitly call another method from the same class
3) Calling constructor of same class - Enables constructor chaining using this().
4) Passing current object as Argument - Pases the current instance to method or another constructor
5) Returning current Instance - Returns the current object from a method.

Q4] Explain the Concept of finalize() method?
→ This method is similar as a destructor in C++ when garbage collector collects the memory of object which are not in use.
Then it will internally calls the finalize method.

The garbage collector thread scanners full memory ~~pasiat~~ partially but we can explicitly call the garbage by using system.gc().

- The finalize() method in java is used to perform cleanup operations before an object is garbage collected.
- It is defined in the object class and can be overridden in a class to release resources like file handles, database connections or network sockets before the object is destroyed.

Syntax-

protected void finalize() throws Throwable

Q5] How can we call the Garbage Collector explicitly in java?

→ In java, garbage collection is automated, meaning the JVM automatically reclaims memory occupied by objects that are no longer referred.

However to run explicitly using the

1] Using System.gc() method.

The System.gc() method suggests that the JVM should run the garbage collector.

However, it does not guarrantee immediate execution

2] Using Runtime.getRuntime().gc()

Another way to request garbage collection is by using the Runtime class.

3] Using finalize() method (Deprecated)

The finalize() method was previously used to perform cleanup before an object is collected by the garabage collector. However, it is deprecated in Java 9 because it is unreliable and may never be called.

Q6] What is difference between finalize() in java and destructor in ~~java~~ C++

→

Finalize() in Java	Destructor in C++
1) Used to perform cleanup before garbage collection	Automatically called when an object is destroyed to free memory and resources.
2) Called by the garbage collector before deleting an object	Called immediately when an object goes out of scope or delete is usede.
3) Can be overridden but not guaranteed	Automatically invoked without manual intervention
4) Defined in object class and can be overriden	A special function with the same name as the class but prefixed with ~.
5) Java's gc handles memory management.	C++ manually manages memo using new and delete.
6) Used for cleanup	Used for deallocating memory and resources explicitly.

Q7] How to display the hashcode of any object in Java

→ In java every obj has a hashcode, which is a unique integer value: generated by the JVM. You can display an object's hashcode using the hashcode() method from the object class.

Using hashCode() Method

The hashcode() method returns an integer that represents the object's memory address

in the JVM. (default implementation)

eg-

```
class Example {}
public class HashCodeDemo {
    public static void main(String[] args) {
        Example obj = new Example();
        System.out.println("Hashcode of obj :" +
                obj.hashCode());
    }
}
```

output : Hashcode of obj : 12345678

Q8] What is alternative for copy constructor in Java.

→ Unlike C++, Java does not have a built-in copy constructor. However there are several alternatives to create clone of an object in Java.

- The clone() method is used to create a shallow copy of an object. It is defined in the object class and must be overridden when used.
- Implement the cloneable interface (to indicate that the object can be cloned).
- Override the clone() method in the class.
- Call super.clone() inside the overridden clone() method.

Q9] What is the use of import statement in Java?

→ The import statement in java is used to access classes without using their fully qualified names.

It makes the code more readable and organized.

Why use import?

- Avoids writting long class names (fully qualified names)
- Allows access to built-in Java libraries (eg. java.util.Scanner).
- Enables modularity by allowing reuse of external or user-defined classes.
- Improves code readability and maintainability.

10] What is difference between static block and constructor?

→

Static block	Constructor
Runs when the class loads	Runs when an obj is created
Executes only once	Executes every time an object is created.
Initializes static variables	Initializes instance variables
Can directly access static members	Cannot modify static members directly.
Called automatically when the class is loaded	called automatically when an object is created
explicit call No, it runs once when the class loads	No, Runs automatically when the object is created.