**HW to Chapters 9 (as in lectures)**

**Classes – OOD, OOP**

Non-programming Assignment:

1. What is class constructor and why is it needed?

A class constructor is a special method used to create and initialize objects in Java. Constructors have the same name as the class and do not have a return type. For example, in a Person class, the constructor might initialize the person's name and age:

public class Person {

private String name;

private int age;

// Constructor for Person class

public Person(String name, int age) {

this.name = name;

this.age = age;

}

}

Constructors are needed to set up initial values for object attributes and to create a new instance of a class.

1. What is the meaning of the following access modifiers: “public”, “private”, “protected”, and “default”?

Access modifiers control the visibility of class members:

"public": Members are accessible from any other class. For example, public int age; means the age variable can be accessed from any class.

"private": Members are accessible only within the class itself. For example, private String name; means the name variable can only be accessed within its own class.

"protected": Members are accessible within the class, its subclasses, and other classes in the same package. For example, protected int height; can be accessed by subclasses and classes in the same package.

"default" (no modifier): Members are accessible only within classes in the same package. For example, int weight; without any modifier means weight can be accessed by classes in the same package.

1. What is the meaning of the following non-access modifiers: “final” and “abstract”?

Non-access modifiers add further restrictions or behavior:

"final": Prevents modification. For example, final int MAX\_VALUE = 100; means MAX\_VALUE cannot be changed. A final class cannot be subclassed, and a final method cannot be overridden.

"abstract": Indicates that a class or method is incomplete and must be implemented by subclasses. For example, abstract class Animal { abstract void makeSound(); } means any subclass of Animal must implement the makeSound method.

1. What is a Java package?

A Java package is a namespace that organizes classes and interfaces into groups for better modularity and preventing naming conflicts. For example, java.util is a package that contains utility classes like ArrayList and HashMap. Packages help in structuring the code logically and provide access control through visibility modifiers.