Department of Artificial Intelligence and Multimedia Gaming

Object oriented Programming (Spring-2025)

LAB No. 06

By completing these lab tasks, students will gain the following key skills:

- Understanding Inheritance Concepts: Learn how to create parent and child classes, and implement different types of inheritance (single, multilevel, and hierarchical).

Lab Task 1: Implementing Single Inheritance

Objective: Understand and implement single inheritance in Java.

Instructions:

1. Create a base class `Animal` with attributes: `name` and `age`.
2. Implement a method `makeSound()` in the base class.
3. Create a derived class `Dog` that extends `Animal`.
4. Override the `makeSound()` method to print "Barking...".
5. Create a `Main` class to create a `Dog` object and call its methods.

Lab Task 2: Implementing Multilevel Inheritance

Objective: Learn multilevel inheritance.

Instructions:

1. Create a base class `Person` with attributes `name` and `age`, and a method `displayInfo()`.
2. Create a derived class `Student` that extends `Person` and adds `studentID`.
3. Create another derived class `GraduateStudent` that extends `Student` and adds `researchTopic`.
4. Implement the constructors for each class.
5. In the `Main` class, create an object of `GraduateStudent` and display its details.

Lab Task 3: Implementing Hierarchical Inheritance

Objective: Understand hierarchical inheritance where multiple classes inherit from a common parent.

Instructions:

1. Create a base class `Vehicle` with attributes `brand` and `speed`, and a method `showDetails()`.
2. Create two derived classes:

   - `Car` with an additional attribute `numDoors`.
   - `Bike` with an additional attribute `helmetType`.

3. Implement the `showDetails()` method in both derived classes.
4. Create a `Main` class to test both `Car` and `Bike`.

Lab Task 4: Implementing Method Overriding in Inheritance

Objective: Implement method overriding in Java inheritance.

Instructions:

1. Create a base class `Shape` with a method `draw()` that prints "Drawing a shape".
2. Create two derived classes:

   - `Circle` that overrides `draw()` to print "Drawing a Circle".
   - `Rectangle` that overrides `draw()` to print "Drawing a Rectangle".

3. Create a `Main` class, declare a `Shape` reference, and assign it objects of `Circle` and `Rectangle` to demonstrate dynamic method dispatch.