National University of Computer and Emerging Sciences

Laboratory Manual

for

Object Oriented Programming Lab

Course Instructor	Mr. Uzair Naqvi
Lab Instructor(s)	Aqib Zeeshan, Seemab Ayub
Section	BCS-2E
Date	Wednesday, 8 May 2024
Semester	Spring 2024

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

- Polymorphism

1 Question#1:

Define a base class `Animal` with attributes like name, age, and methods like `get_info()`. Create derived classes for each animal type (Dog, Cat, Rabbit) inheriting from `Animal` and potentially adding specific attributes (e.g., Dog: breed). Implement a method `generate_report()` in the base class that calls `get_info()`. Override `generate_report()` in derived classes to include specific details relevant to each animal type.

2 Question#2:

Design a base class `Shape` with attributes like position, size, and color. Define methods like `draw()` and `get_area()` in the base class. Create derived classes for specific shapes (Circle, Rectangle, Triangle) inheriting from `Shape`. Override the `draw()` method in each subclass to implement shape-specific drawing logic. Implement the `get_area()` method in each subclass to calculate the area based on the shape's properties. This allows you to create a generic drawing function that calls `draw()` for each shape object, ensuring the appropriate drawing behavior based on the object type. Similarly, the area calculation can be handled polymorphically.

3 Question#3:

Create a base class `Product` with attributes like name, price, and inventory. Define methods like `get_details()` in the base class. Design derived classes for specific product types (Book, Electronic, Clothing) inheriting from `Product`. Override the `get_details()` method in each subclass to include product-specific information. Implement a `calculate_shipping_cost()` method in the base class with a placeholder implementation (e.g., flat rate). Override this method in derived classes to define specific shipping cost calculation logic based on product type (e.g., weight-based for books, flat rate for clothes). This allows you to display product details generically but handle shipping costs differently for each product type.