CSE 112 : Object Oriented Programming Lab
Lab - 10

Intake 52
Section - 03

May 2, 2024

Lab Tasks

Task 1

- **Base Class - Shape:**
  - Create a base class named `Shape` with two double-type properties (`side1` and `side2`).
  - Implement a parameterized constructor in `Shape` to set the values of these properties.
  - Define a member function named `display_area()` in `Shape` to compute and display the area of figures.
- **Derived Class - Triangle:**
  - Derive a specific class called `Triangle` from the base class `Shape`.
  - Implement a constructor in `Triangle` to initialize the properties of the triangle.
  - Override the `display_area()` function in `Triangle` to compute and display the area of the triangle.
- **Derived Class - Rectangle:**
  - Derive another specific class called `Rectangle` from the base class `Shape`.
  - Implement a constructor in `Rectangle` to initialize the properties of the rectangle.
  - Override the `display_area()` function in `Rectangle` to compute and display the area of the rectangle.
- **Runtime Polymorphism:**
  - Utilize runtime polymorphism by creating a pointer of type `Shape` to refer to objects of the derived classes.
  - Assign the address of a `Triangle` object to the `Shape` pointer and use it to display the area of the triangle.
  - Assign the address of a `Rectangle` object to the `Shape` pointer and use it to display the area of the rectangle.

Task 2

- Create an abstract base class `Animal` with a pure virtual function `makeSound()`.
- Derive two classes, `Dog` and `Cat`, from the base class.
- Implement the `makeSound()` function which prints "bark" in `Dog` and "meow" in `Cat`.
- Create an object of the `Animal` class in the `main()` function to call the makeSound() function of the base class.

Task 3

- **Class Design - Flower:**
  - Design a class named `Flower`.
  - The `Flower` class has a single function named `showItem()`.
  - The purpose of `showItem()` is to output what the flower sells.
- **Derived Class - Rose:**
  - Create a derived class named `Rose` from the base class `Flower`.
  - Implement the `showItem()` function in `Rose` to output "sells rose."
- **Derived Class - Marigold:**
  - Create another derived class named `Marigold` from the base class `Flower`.
  - Implement the `showItem()` function in `Marigold` to output "sells marigold."
- **Abstraction Implementation:**
  - Utilize the `Flower` class in the `main()` function.
  - Demonstrate the idea of abstraction by using pointers of type `Flower` to refer to objects of the derived classes (`Rose` and `Marigold`).
  - Use these pointers to call the `showItem()` function, letting the actual implementation details be hidden behind the abstraction.

Task 4

- Write a generic function named `findMaximum` that takes an array of the same data type and returns the maximum value.
  - The generic function is designed to work with arrays of any data type.
  - It iterates through the array to find and return the maximum value.
- In the `main()` function, use the generic function to find the maximum of arrays containing integers, doubles, and characters.
  - For integers: `intArray[] = {5, 10, 3, 8, 2}`
  - For doubles: `doubleArray[] = {3.14, 2.718, 1.618, 2.22, 0.99}`
  - For characters: `charArray[] = {'A', 'B', 'Z', 'D', 'C'}`
- Display the results to show the maximum values for each array type.