CSE 1202 (Object Oriented Programming Lab)

Polymorphism and Exception Handling

1. Define a base class Shape with a virtual function area(). Create derived classes Circle and Rectangle that override the area() function.

   - Circle should calculate the area based on radius.
   - Rectangle should calculate the area based on width and height.

   Write a program that creates objects of Circle and Rectangle, calls their area() function, and displays the calculated areas.

2. Design a base class Vehicle with a virtual function horn(), which outputs "Generic vehicle horn". Create two derived classes Car and Truck.

   - Car should override horn() to output "Car horn: Beep Beep!".
   - Truck should override horn() to output "Truck horn: Honk Honk!".

   Create objects of Car and Truck and demonstrate function overriding by calling the horn() function on each.

3. Write a program that performs division between two numbers entered by the user. If the user tries to divide by zero, throw an exception and catch it to display an error message like "Division by zero is not allowed." Test this with different inputs, including zero as the divisor.

4. Create a program that initializes an array of 5 integers. Ask the user to enter an index to access an element in the array. If the user enters an index out of bounds, throw an exception and catch it to display a message like "Array index out of bounds." Test this by entering indices within and outside the array bounds.

5. Write a program that prompts the user to enter their age. If the user enters an age less than 0 or greater than 150, throw an exception and catch it to display a message like "Invalid age entered." Test this with valid and invalid age inputs to ensure correct handling.

6. Create a BankAccount class with a balance attribute and a withdraw function. In the withdraw function, throw an exception if the withdrawal amount exceeds the balance, with a message like "Insufficient funds." Write a program to create a BankAccount object, deposit some money, and try to withdraw an amount greater than the balance to test the exception handling.