**OBJECT-ORIENTED PROGRAMMING**

|  |  |
| --- | --- |
| Lab 3 | |
| **Topic** | Class, object and getter, setter |
| **Objective** | To practice creating a C++ program with basic OOP structure |

## Task 1:

Create a C++ program for a basic calculator using Object-Oriented Programming (OOP) principles.

**Instructions:**

1. Define a `Calculator` class in a header file (e.g., “**calculator.h”)** with methods for addition, subtraction, multiplication, division, square, cube, square root, cube root, and log.
2. Create the class object in the main program (e.g., **“main.cpp”)** to interact with the `Calculator` class.

In the main program, provide a user menu for selecting an operation. Prompt the user for two numbers as operands.

Use the `Calculator` class methods to perform the selected operation and display the result.

Implement basic error handling, such as division by zero checks.

## Task 2:

Create a C++ program to manage student records using OOP. Define a “**Student”** class with attributes like name, ID, and grades.

**Instructions:**

1. Define a **Student** class with attributes for name, student ID, and an array of grades.
2. Write getter and setter for accessing and updating the attributes.
3. Implement methods to calculate the average grade and display student information.
4. Create the class object in the main program that allows the user to add, edit, delete, or search for student records.
5. Implement error handling for invalid inputs and ensure that student records are managed effectively.

## Task 3:

Create a C++ program to manage an inventory system using OOP. Define a **Product** class with attributes like name, price, and quantity in stock.

**Instructions:**

1. Define a **Product** class with attributes for product name, price, and quantity in stock.
2. Write getter and setter for accessing and updating class attributes.
3. Implement methods for adding, removing, and updating product information, as well as calculating the total inventory value.
4. Create a separate **main program** file that allows the user to interact with the inventory system, including adding, removing, or updating products.
5. Implement error handling for invalid inputs and ensure the inventory is managed effectively.

## Task 4:

Create a C++ program that models basic shapes using OOP. Define classes for circles, rectangles, and triangles, each with methods to calculate area and perimeter.

**Instructions:**

1. Define separate classes for **Circle**, **Rectangle**, and **Triangle**, each encapsulating its respective shape with attributes for length, width, etc.
2. Write getter and setter for accessing and updating the attributes.
3. Implement methods in these classes to calculate the area and perimeter for each shape.
4. Create the classes’ objects in the main program that allow the user to select a shape, input necessary parameters (e.g., radius for a circle, length, and width for a rectangle), and display area and perimeter.
5. Implement error handling for invalid inputs and ensure the correct formula is used for each shape.