**OBJECT-ORIENTED PROGRAMMING**

|  |  |
| --- | --- |
| Graded Lab 1 | |
| **Topic** | Class, object and getter, setter |
| **Objective** | To Evaluate a C++ program with a basic OOP structure |

**Task 1:**

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

**Attributes:**

**length (double):** Represents the length of the swimming pool (in feet).

**width (double):** Represents the width of the swimming pool (in feet).

**depth (double):** Represents the depth of the swimming pool (in feet).

**fillRate (double):** Represents the rate (in gallons per minute) at which the water is filling the pool.

**drainRate (double):** This represents the rate (in gallons per minute) at which the water is draining from the pool.

**Methods:**

Getter methods for each attribute (i.e., **getLength(), getWidth(), getDepth(), getFillRate(), getDrainRate()** ).

Setter methods for each attribute (i.e., **setLength(), setWidth(), setDepth(), setFillRate(), setDrainRate()**).

Complete\_Fill(), partial\_fill: determine the amount of water needed to fill an empty or partially filled pool;

Complete\_drain, partial\_drain: determine the time needed to completely or partially fill or empty the pool; add or drain water for a specific amount of time.

**Note:** use appropriate parameters for filling and draining the pool.

**Main Program (Separate File):**

1. Create instances of the `SwimmingPool` class using both the default and parameterized constructors.

2. Use getter and setter methods to modify and retrieve swimming pool attributes.

3. Calculate the time to fill and drain the swimming pool completely.

4. Calculate the time to partially fill and drain the swimming pool

**Ensure proper error handling for cases like invalid inputs**