**OBJECT-ORIENTED PROGRAMMING**

|  |
| --- |
| Lab 9 |

**Task 1: Point Class with Character Pointers**

1. Implement the Point class in "Point.cpp" based on the declaration in "Point.h."
2. In "Point.cpp," include the necessary functions to set and get the coordinates, and display point details.
3. In "main.cpp," create a **main** function to demonstrate the use of the Point class.
4. In the **main** function, create instances of the Point class using both the default constructor and the parameterized constructor.
5. Display the details of the points, update the coordinates of one point, and print the updated details.
6. Ensure that your implementation works without errors and produces the expected output.

|  |
| --- |
| //Point.h:  #include <iostream>  class Point {  private:  double x;  double y;  public:  // Default constructor  Point();  // Parameterized constructor  Point(double xCoord, double yCoord);  // Display function to print point details  void display() const;  // Function to update the coordinates of the point  void updateCoordinates(double newX, double newY);  }; |

**Task 2:**

**Rational Number Class with Operator Overloading and Static Function**

Implement a C++ class named **RationalNumber** to represent rational numbers. The class should include the following features:

1. **Private Member Variables:**
   * **numerator** (int): to store the numerator of the rational number.
   * **denominator** (int): to store the denominator of the rational number.
2. **Public Member Functions:**
   * Default constructor: Initializes the rational number as 0/1.
   * Parameterized constructor: Accepts parameters for the numerator and denominator and initializes the rational number.
   * **display()**: Displays the rational number in the format "numerator/denominator."
3. **Operator Overloading:**
   * **operator+**: Overloaded for rational number addition.
   * **operator\***: Overloaded for rational number multiplication.
   * **operator==**: Overloaded for rational number equality.

Two rational numbers are considered equal if their simplified forms have the same numerator and denominator.

1. **Static Function:**
   * **static int gcd(int a, int b)**: Computes and returns the greatest common divisor of two integers.

**Task:**

* Implement the **RationalNumber** class with the specified features.
* Overload operators for addition, multiplication, and equality of rational numbers.
* Implement the static function to calculate the greatest common divisor.
* Provide a sample program demonstrating the class functionality.