COMP 116: Object Oriented Programming

Lab Exercises

Lab #1 in Week #3

Question #1

Write a C++ program that reads three coefficients a, b and c for quadratic equation and finds whether the solutions are in real or imaginary. ($ax^2 + bx + c = 0$ if b^2-4ac >=0 then the solutions are real.)

Question #2

Write a C++ program that reads ten positive numbers from user and finally prints the largest of all. (use for loop, if condition and function.)

Lab #2 in Week #4

Question #1

Write a C++ program using function (pass by reference) that calculates the values of x and y from the two linear equations.

ax + by = m

cx + dy = n

The solutions are given as

x = (md - bn)/(ad - cb)

y = (na - mc)/(ad - cb)

The function should take eight arguments and return nothing.

Lab #3 in Week #5

Question #1

Do you remember a graph paper; plotting x-axis, y-axis and origin (0,0). A Point consists of two values; one is x-axis value and other one is y-axis value. Considering only a first quadrant and two such points, write a program that finds the distance between each other. Use classes and objects.

Lab #4 in Week #6

Question #1

Define a class called �`Rectangle`� with following attributes: length and breadth of data type Integer.

Also include the following member functions:

```
void setSize(int length, int breadth); // this function should
set the value of length and breadth of the Rectangle.

int getArea( ); // this function should return the area of the
rectangle.

int getPerimeter( ); // this function should return the
perimeter of the rectangle.
```

// formula to calculate: area = length * breadth.

// formula to calculate: perimeter = 2 * (length + breadth).

Write a driven program as well.

Lab #5 in Week #7 and Week #8

Question #1

`class Complex`

`Private:`
`int x;`
`int y;`

`public:`
`Complex( );`
`Complex( int x, int y);`

Define a class called Complex.

Define member functions that overload the following operators:

- ☒ Minus unary operator. � Returns void
- ☒ Scalar multiplication. (you may use friend function) and returns Complex
- ☒ Plus binary operator (+). => Returns Complex
- ☒ Minus binary operator. => Returns Complex
- ☒ += Shorthand operator. => Returns void
- ☒ == Equals to operator. => Returns TRUE or FALSE
- ☒ Greater than operator. => Returns TRUE or FALSE
- ☒ ! = Not equals to operator. => Returns TRUE or FALSE
- ☒ Pre Increment operator. => Returns Complex
- ☒ Post Increment operator. => Returns Complex
- ☒ << Stream Insertion operator. (use friend function. Why?????) => Returns ostream&

Write a main( ) function to implement the above overloaded operators.

Lab #6 in Week#9 and Week #10

Question #1

Create a base class called *Shape*. Use this class to store two double type values that could be used to compute the area of figures, Derive two specific classes called *Triangle* and *Rectanlge* from the base *Shape*. Add to the base class, a member function *set_data()* to initialize base class data members and another member function *display_data()* to compute and display the area of figures. Make *display_area()* as a virtual function and redefine this function in the derived classes to suit their requirements.

Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.

Class Shape	
public side_one	:double
public side_two	:double
public shape()	:Constructor
public shape(double, double)	:Constructor
public set_data(double, double)	:void
public virtual display_area()	:void

Class Rectangle : Shape	
public display_area()	:void

Class Triangle : Shape	
public display_area()	:void

Lab #7 in Week #11 and Week #12

Question #1

class Shape

protected:
 	
int x;
 	
int y;

public:
 	
Shape( ){ }
 	
Shape(int, int);
 	
virtual void draw ( ) = 0;

Class Circle : public Shape
protected: int radius;

public:
 	
Circle( ){ }
 	
Circle(int, int, int);
 	
void draw( );

Class Ellipse : public Shape

Protected:
int xradius;
int yradius;

public:
Ellispse( ) { }
Ellipse(int, int, int, int);
void draw( );

class Rectangle : public Shape

protected:
int x1;
int y1;

public:
Rectangle( ) { }
Rectangle(int, int, int, int);
void draw( );

Define the above classes and save the file as shape.h

Write a C++ program that displays the following menu.

1. Circle

2. Ellipse

3. Rectangle

4. Exit

Enter your option:

After entering the option, user then enters the position and size of corresponding shape. Now, your program should dynamically (use pointer) allocate the memory and draw that shape. (Use graphics if possible.)

Question #2

Write a function template �*sort(array[ ])*� that sorts the given list of numbers. Write a program that inputs, sorts and outputs an *int* array and a *float* array.

```
// algorithm for sorting

for(m=0; m<size-1; m++)

{

      for(n=m+1; n<size; n++)

      {

            if(array[m]>array[n])    //   for ascending order

            {

                     T   temp;

                     temp = array[m];

                     array[m] = array[n];

                     array[n] = temp;

            }

      }

}

// end of the sorting algorithm.
```

Lab #8 in Week #13

Question #1

Write a program that illustrates the concept of Exceptional Handling in C++.