Homework #6

Turn In:

1. Exercise #1 – Due in class on Tuesday, November 8, 2016
   a) For each exercise, a hardcopy package must be generated to include the following items:
      - Cover Sheet (use the sample copy include in class/lecture note)
      - Exercise/problem statement
      - Copy of your source files (for examples, **`cis25Fall2016YournameHw6Ex1.cpp`**, **`fractionYourName.h`**, **`fractionYourName.cpp`**, **`fractionUtilityYourName.h`**, **`fractionUtilityYourName.cpp`**, **`pointYourName.h`**, **`pointYourName.cpp`**, **`pointUtilityYourName.h`**, **`pointUtilityYourName.cpp`**, etc.)
      - Copy of output (copy and paste to the end of your application program as COMMENT block)
      - Copy of YOUR COMMENTS (as a separate comment block) after YOUR PROGRAM OUTPUT
   b) Submitting in class one hard copy for each document
   c) Emailing each document as follows,
      - One message for each exercise.
      - Attaching the source file that was created in part a).
      - The SUBJECT line of the message should have the following line:

        `cis25Fall2016YourNameHw6Ex1`

3. Q.E.D.

1. Code Assignment/Exercise

EXERCISE 1

Consider the following classes:

```
class FractionYourName;

class PointYourName; // To Be Created
```

The incomplete class definitions and code are given as follows (and to be updated as given in class discussions),

```
// Header Files

/**
 * Program Name: fractionYourName.h
 * Discussion:   Declaration File --
 *                 FractionYourName class
 */
#ifndef FRACTIONYOURNAME_H
#define FRACTIONYOURNAME_H

class FractionYourName {
public:

  // YOUR CODE HERE
  //   Must have at least the default constructor,
  //                          copy contructor,
  //                          destructor, and
  //                          assignment operator function
  //   and other members

private:
  int num; // numerator will preserve fraction-negativity;
           // i.e., negativity of a fraction will be
           // assigned to its numerator

  int denom; // non-zero value for denominator
};

// your I/O OPERATOR functions here

#endif

/**
 * Program Name: pointYourName.h
 * Discussion:   Declaration File --
 *                 PointYourName Class
 */
#ifndef POINTYOURNAME_H
#define POINTYOURNAME_H

#include "fractionYourName.h"

// Declarations
```

```
class PointYourName {
public:

  // YOUR CODE HERE
  //   Must have at least the default constructor,
  //                          copy contructor,
  //                          destructor, and
  //                          assignment operator function

  // operations

  int getQuadrant() {
    // YOUR CODE HERE
  }

  int getQuadrant(const PointYourName& p) {
    // YOUR CODE HERE
  }

  void moveBy(FractionYourName delX, FractionYourName delY) {
    // YOUR CODE HERE
  }

  void moveBy(int iOld) {  // update as needed
    // YOUR CODE HERE
  }

  void flipByX() {  // update as needed
    // YOUR CODE HERE
  }

  void flipByY() {  // update as needed
    // YOUR CODE HERE
  }

  void flipThroughOrigin() {  // update as needed
    // YOUR CODE HERE
  }

  void print() {  // update as needed
    // YOUR CODE HERE
  }

  // add operator functions as needed

private:
  FractionYourName x; // x-coordinate of the point
  FractionYourName y; // y-coordinate of the point
};

// your I/O OPERATOR functions here

#endif
```

You are asked to

(1) Add more member functions and operator functions as needed for the **`Point`** class; and

(2) Provide complete definitions for all member functions so that the given class is proper and working properly; and

(3) Add/Provide complete definitions for all needed non-member functions to perform reasonable tasks; and

(4) Save all classes in appropriate `*.h` and `*.cpp` files with appropriate names; and

(5) Run a menu program named as with a driver named as **cis25Fall2016YourNameHW6Ex1Driver.cpp** and save the output. A sample program output is given as follows,

(a) The output screen should have the following lines displayed before any other display or input can be seen,

```
CIS 25 – C++ Programming
Laney College
Your Name

Assignment Information --
  Assignment Number:  Homework 06,
                      Exercise #1
  Written by:         Your Name
  Due Date:           Due Date
```

(b) Then, the output screen should also be followed by,

```
*******************************
*         MENU – Hw #6        *
*  1. Initializing (2 Points) *
*  2. Placement               *
*  3. Moving                  *
*  4. Flipping                *
*  5. Displaying              *
*  6. Quit                    *
*******************************
Select an option (use integer value only): 2

    Not a proper call as no Points are available!

*******************************
*         MENU – Hw #6        *
*  1. Initializing (2 Points) *
*  2. Placement               *
*  3. Moving                  *
*  4. Flipping                *
*  5. Displaying              *
*  6. Quit                    *
*******************************
Select an option (use integer value only): 1

  Initializing Option --

    // Providing proper values & steps!

*******************************
*         MENU – Hw #6        *
*  1. Initializing (2 Points) *
*  2. Placement               *
*  3. Moving                  *
*  4. Flipping                *
*  5. Displaying              *
*  6. Quit                    *
*******************************
Select an option (use integer value only): 5

  Printing Option --

    // Displaying proper values & formats!

*******************************
*         MENU - Hw #6        *
*  1. Initializing (2 Points) *
*  2. Placement               *
*  3. Moving                  *
*  4. Flipping                *
*  5. Displaying              *
*  6. Quit                    *
*******************************
Select an option (use integer value only): 3

  Moving Option --

    *****************************
    *  Sub MENU -- MovingPoint  *
    *  1. By (frX, frY)         *
    *  2. By fr                 *
    *  3. Printing              *
    *  4. Returning             *
    *****************************
    Select an option (use integer value only): 1

      // Providing proper values & steps!

    *****************************
    *  Sub MENU -- MovingPoint  *
    *  1. By (frX, frY)         *
    *  2. By fr                 *
    *  3. Printing              *
    *  4. Returning             *
    *****************************
    Select an option (use integer value only): 2

      // Providing proper values & steps!

    *****************************
    *  Sub MENU -- MovingPoint  *
    *  1. By (frX, frY)         *
    *  2. By fr                 *
    *  3. Printing              *
    *  4. Returning             *
    *****************************
    Select an option (use integer value only): 3
```

```
      // Displaying proper values & formats!

    *****************************
    *  Sub MENU -- MovingPoint  *
    *  1. By (frX, frY)         *
    *  2. By fr                 *
    *  3. Printing              *
    *  4. Returning             *
    *****************************
    Select an option (use integer value only): 4

    Returning to “MENU – Hw#6”

*******************************
*         MENU – Hw #6        *
*  1. Initializing (2 Points) *
*  2. Placement               *
*  3. Moving                  *
*  4. Flipping                *
*  5. Displaying              *
*  6. Quit                    *
*******************************
Select an option (use integer value only): 4

  Flipping Option --

    ****************************
    * Sub MENU - FlippingPoint *
    *  1. By Y                 *
    *  2. By X                 *
    *  3. By Origin            *
    *  4. Printing             *
    *  5. Returning            *
    ****************************
    Select an option (use integer value only): 1

      // Providing proper values & steps!

    ****************************
    * Sub MENU - FlippingPoint *
    *  1. By Y                 *
    *  2. By X                 *
    *  3. By Origin            *
    *  4. Printing             *
    *  5. Returning            *
    ****************************
    Select an option (use integer value only): 2

      // Providing proper values & steps!

    ****************************
    * Sub MENU - FlippingPoint *
    *  1. By Y                 *
    *  2. By X                 *
    *  3. By Origin            *
    *  4. Printing             *
    *  5. Returning            *
    ****************************
    Select an option (use integer value only): 3

      // Providing proper values & steps!

    ****************************
    * Sub MENU - FlippingPoint *
    *  1. By Y                 *
    *  2. By X                 *
    *  3. By Origin            *
    *  4. Printing             *
    *  5. Returning            *
    ****************************
    Select an option (use integer value only): 4

      // Displaying proper values & formats!

    ****************************
    * Sub MENU - FlippingPoint *
    *  1. By Y                 *
    *  2. By X                 *
    *  3. By Origin            *
    *  4. Printing             *
    *  5. Returning            *
    ****************************
    Select an option (use integer value only): 5

    Returning to “MENU – Hw #6”

*******************************
*         MENU – Hw #6        *
*  1. Initializing (2 Points) *
*  2. Placement               *
*  3. Moving                  *
*  4. Flipping                *
*  5. Displaying              *
*  6. Quit                    *
*******************************
Select an option (use integer value only): 6

  Having Fun ...
```

Note!

You should at least test your program with the information given below.

Point #1: (1/2, 2/1)

Point #2: (4/1, 1/1)

Point #3: (-1/1, -1/2)

Point #4: (2/1, -2/1)