Homework 2 (due 1/30)

1. (20 pts.) Implement a class named `NamedSquare` to represent named squares with real coordinates on the plane. Your class must be **derived** from `Rectangle` and must support the following member functions:

   1.
      ```
      NamedSquare(Point c = Point(), double s = double(), std::string n =
      std::string());

      // pre-condition: s is nonnegative

      // post-condition: creates new NamedSquare centered at c with side s and

      //                 name n (this is the default constructor)

      // NOTE: since NamedSquare has no access to the private variables of

      //       Rectangle, use the following syntax in your implementation:

      //       NamedSquare::NamedSquare(Point c, double s, std::string n):
      Rectangle(c, s, s), _name(n) {}
      ```

   2.
      ```
      NamedSquare(const NamedSquare & source);

      // pre-condition: none

      // post-condition: creates a new NamedSquare that is a copy of the given

      //                 NamedSquare source (this is the copy constructor)

      // NOTE: the parameter MUST be passed as specified (constant reference)

      //       to avoid infinite recursion

      // NOTE: since NamedSquare has no access the private variables of

      //       Rectangle, use the following syntax in your implementation:

      //       NamedSquare::NamedSquare(const NamedSquare & source):
      Rectangle(source.Rectangle::get_center(), source.get_side(),
      source.get_side()), _name(source._name) {}
      ```

   3.
      ```
      double get_side() const;

      // pre-condition: none

      // post-condition: returns the side of this NamedSquare
      ```

   4.
      ```
      std::string get_name() const;

      // pre-condition: none

      // post-condition: returns the name of this NamedSquare
      ```

   5.
      ```
      void set_side(double s);

      // pre-condition: s is nonnegative
      ```

```
// post-condition: s is the new side of this NamedSquare
```

6. `void set_name(std::string n);`

   ```
   // pre-condition: none
   // post-condition: n is the new name of this NamedSquare
   ```

7.
   ```
   double get_length() const  = delete;
   double get_width() const = delete;
   void set_length(double) = delete;
   void set_width(double) = delete;
   // the above member functions inherited from Rectangle are deleted
   // they are replaced by get_side() and set_side()


   // NOTE: get_center(), get_area(), get_perimeter(), get_lower_left(),
   // get_upper_right(), set_center() are still available but they must be
   // preceded by Rectangle:: (i.e., Rectangle::get_center(), etc.) if used in
   // a member function of NamedSquare
   ```

Furthermore, the following operators are overloaded with a *nonmember* function:

1. `std::ostream & operator << (std::ostream & os, NamedSquare ns);`

   ```
   // pre-condition: none
   // post-condition: NamedSquare ns is output to stream os in the following
   //                 format: [(lower left), (upper right), name]
   ```

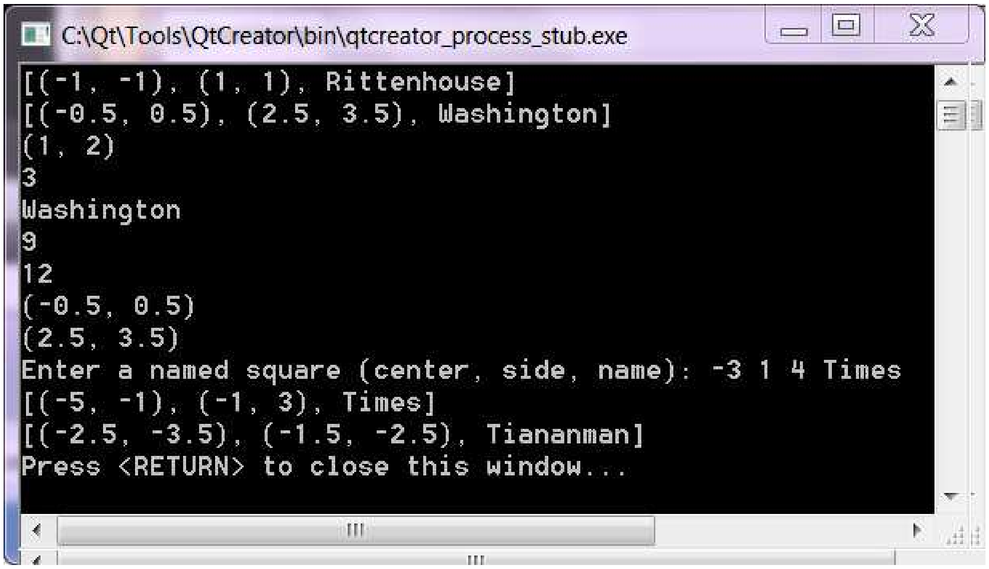
2. `std::istream & operator >> (std::istream & is, NamedSquare & ns);`

   ```
   // pre-condition: none
   // post-condition: a new value of NamedSquare ns is read from stream is in
   //                 the following format: center side name
   ```

Use the following representation for `NamedSquare`:

```
std::string _name;
// INVARIANTS: _name is the name of this NamedSquare
//
// NOTE: private variables of Rectangle are NOT accessible to NamedSquare member
//       functions.
//       To retrieve their values, use Rectangle::get_center(), ...
//       To set their values, use Rectangle::set_center(c) or the constructors
//                                Rectangle(c, l, w) or Rectangle(ll, ur)
```

The output of the test `main.cpp` (downloadable from the homework directory) should be:



C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe

```
[(-1, -1), (1, 1), Rittenhouse]
[(-0.5, 0.5), (2.5, 3.5), Washington]
(1, 2)
3
Washington
9
12
(-0.5, 0.5)
(2.5, 3.5)
Enter a named square (center, side, name): -3 1 4 Times
[(-5, -1), (-1, 3), Times]
[(-2.5, -3.5), (-1.5, -2.5), Tiananman]
Press <RETURN> to close this window...
```

Figure 1. Output of `main-hw2.cpp` (homework directory)

You must turn in by noon of the due date:

- a hard copy of your code with your name, section number, and sample output; and
- send **one** email message with the subject line: `HW2 Your_last_name Your_section` to `cs60@math.scu.edu` with your code attached. Please do **not** zip your file.