C++ Inheritance

Due back on Saturday, March 25 before 11:59 pm.

Goal

You are asked to create a commandline-based interactive program that allows users to add two types of shapes: triangles and circles. The program also allows the user the print the list of stored shapes. Furthermore, it allows the user to modify the stored shapes.

Sample run of this program

The following run illustrates the interactions supported by this program.

./lab7
Choose a shape to add
  (a) for triangle
  (b) for circle
  (p) to print the shapes
  (0-9) to modify a shape
  (q) to exit

: a
== You selected a triangle. ==
Enter position: 0 0
Enter width and height: 1 2
Choose a shape to add
  (a) for triangle
  (b) for circle
  (p) to print the shapes
  (0-9) to modify a shape
  (q) to exit

: b
== You selected a circle. ==
Enter position: 10 2
Enter radius: 1
Choose a shape to add
  (a) for triangle
  (b) for circle
  (p) to print the shapes
  (0-9) to modify a shape
  (q) to exit

: a
== You selected a triangle. ==
Enter position: 3 4
Enter width and height: 1 22
Choose a shape to add
  (a) for triangle
  (b) for circle
  (p) to print the shapes
  (0-9) to modify a shape
  (q) to exit

: p
[0] Triangle
  location: (0, 0)
  width and height: (1, 2)
[1] Radius
  location: (10, 2)
  radius: (1)
[2] Triangle
  location: (3, 4)
  width and height: (1, 22)
Choose a shape to add
  (a) for triangle
  (b) for circle
  (p) to print the shapes
  (0-9) to modify a shape
  (q) to exit

: 1
Enter position: 11 3
Enter radius: 1000
Choose a shape to add
  (a) for triangle
  (b) for circle
  (p) to print the shapes
  (0-9) to modify a shape
  (q) to exit

: p
[0] Triangle
  location: (0, 0)
  width and height: (1, 2)
[1] Radius
  location: (11, 3)
  radius: (1000)
[2] Triangle
  location: (3, 4)
  width and height: (1, 22)
Choose a shape to add
  (a) for triangle
  (b) for circle
Some specifics

- If user enters a, the program prompts the user for information about a triangle (position x, position y, width, and height).
- If user enters b, the program prompts the user for information about a circle (position x, position y, radius).
- If user enters p, the program prints the shapes (their types, position x, position y, and other info, e.g., width and height in case of triangles).
- The program can store up to 10 shapes.

- If user presses 0-9 the shape stored at that index allows user to re-enter the values (e.g., position x, position y, and radius for circle).

Source Code

Please complete the Tri and Cir classes in the code below:

```cpp
#include <iostream>
using namespace std;

class Point {
    public:
        int x, y;

    Point() {}
    Point(int x, int y) { this->x = x; this->y = y; }

    friend ostream& operator<<(ostream& os, const Point& pt) {
        os << "(" << pt.x << ", " << pt.y << ")";
        return os;
    }
};

class Shape {
    // FEEL FREE TO CHANGE THE THIS CLASS
    // AS LONG AS YOU DON'T HAVE TO CHANGE THE
    // CONTENTS OF MAIN() AND
    // PRINT_SHAPES() FUNCTIONS

    protected:
        Point _location;

    public:
        Shape() {}
        Shape(const Point& location) : _location(location) {}
        ~Shape() {}
```
void draw() {
    cout << "location: " << _location << endl;
}

void get_info_from_user() {
    cout << "Enter position: ";
    cin >> _location.x >> _location.y;
}

// YOU NEED TO COMPLETE THIS CLASS
class Tri
{
    protected:
    int _width, _height;

    public:
    // TODO ...
};

// YOU NEED TO COMPLETE THIS CLASS
class Cir
{
    protected:
    int _radius;

    public:
    // TODO ...
};

// DO NOT CHANGE CODE BELOW THIS LINE
#define MAX_SHAPES 10

void print_shapes(Shape** shapes, int n)
{
    for (int i=0; i<n; ++i) {
        cout << "[" << i << "]\n" ; shapes[i]->draw();
    }
}

int main()
{
    Shape* shapes[MAX_SHAPES];

    int n = 0;
    char shape_choice;
    do {
        cout << "Choose a shape to add" << endl;
        cout << "\t(a) for triangle" << endl;
        cout << "\t(b) for circle" << endl;
        cout << "\t(p) to print the shapes" << endl;
        cout << "Enter the shape choice: ";
        cin >> shape_choice;

        if (shape_choice == 'a')
            shapes[n++] = new Tri;
        else if (shape_choice == 'b')
            shapes[n++] = new Cir;
        else if (shape_choice == 'p')
            print_shapes(shapes, n);
    } while (shape_choice != 'p');
cout << "t(0-9) to modify a shape" << endl;
cout << "t(q) to exit" << endl;
cout << ": ";
cin >> shape_choice;

if (shape_choice == 'q') break;

switch(shape_choice)
{
    case '0':
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9':
        shapes[((int) shape_choice) - 48]->get_info_from_user();
        break;
    case 'a':
        if (n >= MAX_SHAPES) {
            cout << "Cannot add any more shapes" << endl;
            continue;
        }
        cout << "You selected a triangle." << endl;
        shapes[n] = new Tri();
        shapes[n]->get_info_from_user();
        ++n;
        break;
    case 'b':
        if (n >= MAX_SHAPES) {
            cout << "Cannot add any more shapes" << endl;
            continue;
        }
        cout << "You selected a circle." << endl;
        shapes[n] = new Cir();
        shapes[n]->get_info_from_user();
        ++n;
        break;
    case 'p':
        print_shapes(shapes, n);
        break;
    case 'q':
        break;
    default:
        cerr << "Invalid choice" << endl;
        break;
}
} while (true);

for (int i=0; i<n; ++i) delete shapes[i];
Submission

Please submit via Blackboard