CSE 4252: Lab 5 (20 Points)

Due Date: November 14th 11:59 PM

Overview

This exercise will allow you to have some practice with basic concepts of Inheritance and Run-time Polymorphism in C++.

Exercise 1 Description

classes.cpp

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

class A {
 protected:
  int var;
 public:
  A() { var = 0;}
  void increment() { var++; }
  virtual void decrement() { var--; }
  void print() {cout << "var = " << var << endl;}
};

class B : public A {
 public:
  void increment() { var += 3; }
  virtual void decrement() { var -= 3; }
};

int main() {
  A a;
  A * ptr;
  ptr = &a;
  ptr -> increment();
  ptr -> print();
  ptr -> decrement();
  ptr -> print();
}
```

Compile the program by typing `g++ classes.cpp`, run it by typing `./a.out`. Make sure that the program compiles and runs.

Question 1. In main add an object `b` of class B. Set the pointer `ptr` to point to b. Add a protected variable `varB` to B and a constructor to initialize it to 0. Add a `print()` method in B so that both `var` and `varB` are printed. Which print method is called on the object B referenced by `ptr`? If needed, change method declarations so that **the print method of B is called for the B object**. What did you need to change and why?

Question 2. Assuming that `a` is a variable of type A, `b` is an object of type B, and `ptr` is the pointer of type A, what would you expect to be printed by the following statement:

```
a = b;
a.print();
ptr = &a;
ptr -> print();
```

What gets printed? Why? Explain the difference in behavior between this example and Question 1.

Question 3. What happens if you have an object b of class B and you call the print method like this: `b.A::print()?`

Question 4. Write a function that takes an object of class A and returns it. Can you pass an object of a class B to this function?

Question 5. What happens if you change the declaration

`class B : public A`
to
`class B : A` ?

You should submit the modified **classes.cpp** file and **Answers.txt (**should show the output of each Question**)** with your answers/observations to each question. Record the *script* in **Lab5_1.txt**.

Exercise 2 Description

Implement the following hierarchy using C++ classes.

1. *Shape:* This should be an **interface** class that supports area() and display() functions.
2. *Circle and Rectangle:* Implement the area and display functions of Shape.
3. *Quadrilateral:* This class should be an **abstract** class. It may contain member variables for length and height.
4. *Trapezoid:* Should inherit length, height from Quadrilateral, and add 'side', which represents the side parallel to 'length'.
5. Create appropriate .h/.cpp files with constructors and destructors (wherever needed).

There is no mandatory requirement on how to plan your .h and .cpp files.

Your **main.cpp** should look like:

```
void print(const Shape& s) {
    s.display();
}

int main() {
    Circle c(10);
    Rectangle r(3, 4);
    //Quadrilateral q(2, 4); //Uncommenting this line should cause an error

    Trapezoid trap(3, 4, 5);

    print(c);
    print(r);
    print(trap);

    return 0;
}
```

Output

```
Circle: 10 | Area = 314.159
Rectangle: 3, 4 | Area = 12
Trapezoid: 3, 4, 5 | Area = 16
```

You should submit a **Readme** file with instructions on how to run your code. Test the code and record the *script* in **Lab5_2.txt**.

Submission Instructions

Make sure your programs compile and run correctly before submitting. To submit, zip all the files in **Lab05.zip** and upload on carmen. Mandatory files:
classes.cpp: [3 points]
Answers.txt: [5 points]
Lab5_1.txt: [1 point]

.h/.cpp files: [5 points]
main.cpp: [2 points]
Readme: [2 points]
Lab5_2.txt: [2 points]