Creative Software Design, Assignment 7-1

Deadline: 2024-10-16 23:59 (No score for late submission)

- Submit your homework by uploading your zip file to the LMS assignment section. Below is an example.

```
13178_Assignment1-1_2024123456.zip
├─ 1.cc
├─ 2.cc
├─ 3.cc
└─ ...
```

- Your zip file name should follow this format:
  13178_Assignment[Assignment-number]_[Student-ID].zip
  - Ex. 13178_Assignment1-1_2024123456.zip
- Source files should be named as **<filename>.cc** *or* **<filename>.cpp**
- **You must submit your solution in the zip file before the deadline.**

1. **Write a C++ Program for Adding and Subtracting Two Integers.**

   A. **Create `Adder` and `Subtractor` classes that perform addition and subtraction of two integers.**

      1. The `Adder` and `Subtractor` classes should inherit from the `Calculator` class.

   B. **You are given the following `Calculator` class definition, which should not be modified.**

```
class Calculator {
    void input(){
        cout << "Input 2 integers >> ";
        cin >> a >> b;
    }
protected:
    int a, b;
    virtual int calc(int a, int b) { return 0; }
public:
    void run() {
        input();
        cout << "Answer: " << calc(a,b) << endl;
    }
};
```

   C. **Example of the `main()` function.**

```
int main() {
    Calculator *adder = new Adder;
    Calculator *subtractor = new Subtractor;
    adder->run();
    subtractor->run();
}
```

   D. Example output of your program (Bold text indicates user input):

```
Input 2 integers >>2 4⏎
Answer: 6
Input 2 integers >>5 7⏎
Answer: -2
```

   E. Submission file: one C++ source file (File name: 1.**cc** or 1.**cpp**)

2. Write a C++ Program for Converting Kilometers to Miles.

A. Create a `KmToMile` class that converts kilometers to miles:

1. The `KmToMile` class should inherit from the `Converter` class and should not have any member variables.

B. The `Converter` class is defined as follows. You should not modify the Converter class.

```
class Converter {
protected:
    double _ratio;

    virtual double convert(double src) { return 0; }
    virtual string getSrcMetric() { return ""; }
    virtual string getDestMetric() { return ""; }

public:
    Converter(double ratio) : _ratio(ratio) {}

    double getRatio() const { return _ratio; }

    void run() {
        double src;
        cout << "Convert " << getSrcMetric() << " to "
             << getDestMetric() << endl;
        cout << "Input " << getSrcMetric() << ": ";
        cin >> src;
        cout << "Result: " << convert(src) << " "
             << getDestMetric() << endl;
    }
};
```

C. Example of the `main()` function.

```
int main() {
    Converter *ktm = new KmToMile(0.621371);
    ktm->run();
}
```

D. Example output of your program (Bold text indicates user input):

```
Convert km to mile
Input km: 25↵
Result: 15.5343 mile
```

E. Submission file: one C++ source file (File name: 2.**cc** or 2.**cpp**)

3. Write a C++ Program to Calculate the Areas of an Ellipse, Rectangle, and Triangle.

A. Create `Ellipse`, `Rect`, and `Triangle` classes that calculate the areas of the corresponding shapes.

1. These classes should inherit from the `Shape` class, and no new member functions should be added for the three classes.

B. The `Shape` class is defined as follows. You should not modify the Shape class:

```
class Shape {
    string name;
protected:
    int width, height;

public:
    Shape(string n, int w, int h)
        : name(n), width(w), height(h) {}

    virtual double getArea() { return 0; }

    string getName() const { return name; }
};
```

C. Example of the `main()` function:

```
int main() {
    vector<Shape*> p(3);
    p[0] = new Ellipse("Ellipse", 10, 20);
    p[1] = new Rect("Rectangle", 30, 40);
    p[2] = new Triangle("Triangle", 30, 40);

    for (int i = 0; i < 3; i++) {
        cout << p[i]->getName() << "'s area is "
             << p[i]->getArea() << endl;
    }

    for (int i = 0; i < 3; i++)
        delete p[i];

    return 0;
}
```

D. Example output of your program (Bold text indicates user input):

```
Ellipse's area is 628
Rectangle's area is 1200
Triangle's area is 600
```

E. Submission file: one C++ source file (File name: 3.**cc** or 3.**cpp**)