Creative Software Design, Assignment 10-1

Deadline: 2024-10-30 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 Book class with three == comparison operator functions.**

   A. **Define the Book class as follows:**

      1. Implement three == comparison operator functions outside of the class and declare them as friend functions within the `Book` class.

```
class Book {
    string title;
    int price, pages;

public:
    Book(string title = "", int price = 0, int pages = 0) {
        this->title = title;
        this->price = price;
        this->pages = pages;
    }

    void show() {
        cout << "Book: " << title << " " << price << " won " << pages
        << " pages" << endl;
    }

    // Implement friend functions here
};

// Implement comparison operator functions here
```

   B. **Example of the `main()` function:**

```
int main() {
    Book a("Dune1", 30000, 500), b("Dune2", 30000, 500);
    a.show();
    if (a == "Dune1") cout << "This book is Dune1" << endl;
    if (a == 30000) cout << "Original price is 30000 won" << endl;
    if (a == b) cout << "They are the same book" << endl;
}
```

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

```
Book: Dune1 30000 won 500 pages
This book is Dune1
Original price is 30000 won
```

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

2. **Write a Circle class with increment operators.**

   A. **The class should use both prefix and postfix increment operators to increase the radius.**

   B. **Define the Book class as follows:**

      1. Implement prefix and postfix increment operator functions outside of the class and declare them as friend functions within the `Circle` class.

```
class Circle {
    int radius;

public:
    Circle(int radius = 0) { this->radius = radius; }
    void show() { cout << "Circle's radius = " << radius << endl; }

    // Implement friend functions here
};

// Implement increment operator functions here
```

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

```
int main() {
    Circle a(5), b;
    b = ++a;
    a.show();
    b.show();
    b = a++;
    a.show();
    b.show();
}
```

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

```
Circle's radius = 6
Circle's radius = 6
Circle's radius = 7
Circle's radius = 6
```

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

3. Write a SortedArray class that sorts an integer array in increasing order.

A. Define the SortedArray class as follows:

1. Implement the declared member functions, including a copy constructor, + arithmetic operator, and = assignment operator.

```
class SortedArray {
    int size;        // size of array
    int *p;          // pointer
    void sort();     // sort integer array in increasing order

public:
    SortedArray();                // initialize p to NULL, size to 0
    SortedArray(const SortedArray& src);       // copy constructor
    SortedArray(int p[], int size);            // constructor
    ~SortedArray();                            // destructor

    // add op2 array to current array
    SortedArray operator+ (const SortedArray& op2);
    // copy op2 array to current array
    SortedArray& operator= (const SortedArray& op2);
    void show();
};
```

B. **Example of the `main()` function:**

```
int main() {
    int n[] = {2, 300, 68};
    int m[] = {100, 5, 6, 65};
    SortedArray a(n, 3), b(m, 4), c;

    c = a + b;

    a.show();
    b.show();
    c.show();
}
```

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

```
Array: 2 68 300
Array: 5 6 65 100
Array: 2 5 6 65 68 100 300
```

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