Object-Oriented Programming Language

11/14/2018

Homework Assignment No. 7

Due 11:59 pm, Thursday November 22, 2018

Late submission within 24 hours: score*0.9;

Late submission within one week: score*0.8.

The solutions will be posted after one week of the due date.

(Total 50%)

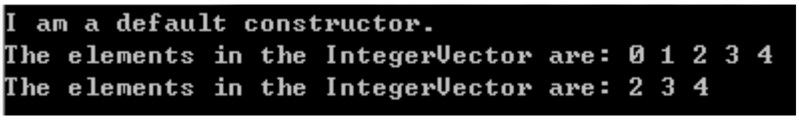
1. **(25%)**: (a) Write an `IntegerVector` class for an array of integers. In the class, use `std::vector` as your internal data representation and provide the following constructors:

- `IntegerVector()` // print a message said “I am a default constructor.”
- `IntegerVector(unsigned nelems)` // creates an `IntegerVector` with the integers `0...nelems-1`
- `IntegerVector(unsigned start, unsigned end)` // creates an `IntegerVector` with the range [start, end)

In addition, provide a non-member `print` function. Test your program with the client code listed below:

```
#include <iostream>
#include "IntegerVector.h"
using namespace std;

int main()
{
    IntegerVector a;
    IntegerVector b(5);
    print(cout, b);
    IntegerVector c(2, 5);
    print(cout, c);
}
```



```
I am a default constructor.
The elements in the IntegerVector are: 0 1 2 3 4
The elements in the IntegerVector are: 2 3 4
```

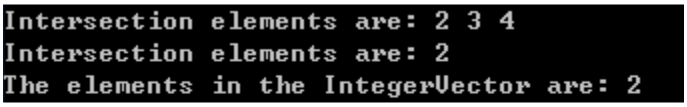
(b) Add an `intersection` member function that prints out the elements common to two arrays. Allow a sequential operation. Test your program with the client code listed below:

```
#include <iostream>
#include "IntegerVector.h"
```

```
using namespace std;

int main()
{
    IntegerVector a(3);
    IntegerVector b(5);
    IntegerVector c(2, 6);
    print(cout, c.intersection(b).intersection(a));
}
```

Your output looks like:



```
Intersection elements are: 2 3 4
Intersection elements are: 2
The elements in the IntegerVector are: 2
```

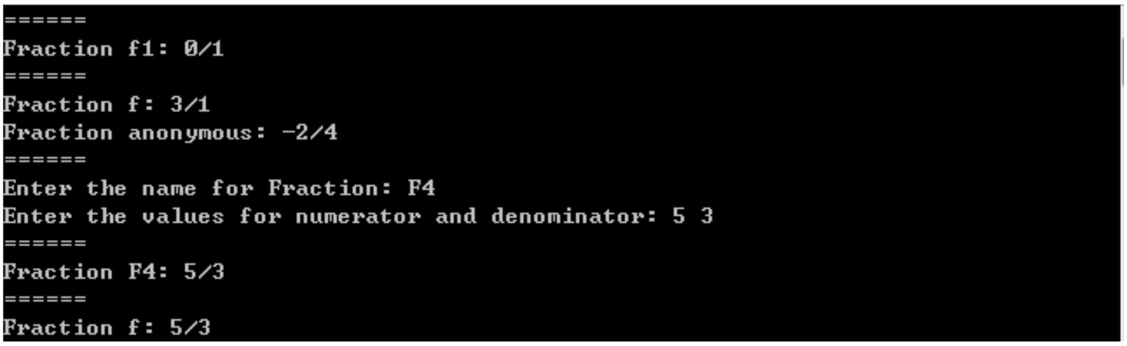
2. **(25%)**: Rational numbers (fractions) are numbers that can be written in the form *a*/*b*, where *a* and *b* are integers and *b* != 0. *a* is known as the *numerator* and *b* the *denominator.* Let the default value for *a* be 0 and for *b* be 1. Implement a class `Fraction` to represent rational numbers and allow their objects to support the following client code:

```
#include <iostream>
#include "Fraction.h"

using namespace std;

int main(){
    Fraction f1; // 0/1
    f1.setName("f1");
    cout << "======" << endl;
    printFraction(cout, f1);
    Fraction f2(3); //3/1
    f2.setName("f");
    Fraction f3(-2, 4); //-2/4
    cout << "======" << endl;
    printFraction(printFraction(cout, f2), f3);
    cout << "======" << endl;
    Fraction f4(cin); // prompt for input
    cout << "======" << endl;
    printFraction(cout, f4);
    cout << "======" << endl;
    printFraction(cout, f4.setName(&f2));
    return 0;
}
```

Below is a sample run:



```
======
Fraction f1: 0/1
======
Fraction f: 3/1
Fraction anonymous: -2/4
======
Enter the name for Fraction: F4
Enter the values for numerator and denominator: 5 3
======
Fraction F4: 5/3
======
Fraction f: 5/3
```