Object-oriented Programming Operator Overloading Part 1 Exercise

YoungWoon Cha

Computer Science and Engineering

KU KONKUK UNIVERSITY

In-class Exercise

In-Class Exercise

Q1 [10 points]. Given the initial *Fraction* class in the slides, extend the class based on the application code in the next page.

Q2 [5 points]. Choose an operator in Table 10.1 other than the ones in Q1. Implement its operator overloading. Show that your operator overloading works correctly with the steps at the end of the application code.

Please implement the entire code in a single cpp file.

Please submit your report in a single PDF file by the end of the class today.

The report should include the submitter's information, the source code, and the output screenshot.

In-Class Exercise: Application Code Part 1

```
int main()
{
   // Creation of two objects and testing the plus and minus operator
   Fraction fract1(2, 3);
   Fraction fract2(1, 2);
   cout << "fract1: " << fract1.print() << endl;
   cout << "fract2: " << fract2.print() << endl;
   +fract1;
   -fract2;
   cout << "Result of +fract1: " << fract1.print() << endl;
   cout << "Result of -fract2: " << fract2.print() << endl << endl;

   // Creation of four objects and testing the ++ and -- operators
   Fraction fract3(3, 4);
   Fraction fract4(4, 5);
   Fraction fract5(5, 6);
   Fraction fract6(6, 7);
   cout << "fract3: " << fract3.print() << endl;
   cout << "fract4: " << fract4.print() << endl;
   cout << "fract5: " << fract5.print() << endl;
   cout << "fract6: " << fract6.print() << endl << endl;
   ++fract3;
   --fract4;
   Fraction fract55 = fract5++;
   Fraction fract66 = fract6--;
   cout << "Result of ++fract3: " << fract3.print() << endl;
   cout << "Result of --fract4: " << fract4.print() << endl;
   cout << "Result of fract5++: " << fract5.print() << endl;
   cout << "Result of fract6--: " << fract6.print() << endl << endl;
```

In-Class Exercise: Application Code Part 2

```
// Testing assignment & inequality operators
if (fract3 != fract4)
{
   fract3 = fract4;
}
cout << "Result of fract3 != fract4: "
     << to_string(fract3 != fract4) << endl;
cout << "fract3: " << fract3.print() << endl << endl;

// Testing compound assignment operators
Fraction fract7(3, 5);
Fraction fract8(4, 7);
Fraction fract9(5, 8);
Fraction fract10(7, 9);

fract7 += 2; // == Fraction(2, 1)
fract8 -= 3; // == Fraction(3, 1)
fract9 *= 4; // == Fraction(4, 1)
fract10 /= 5; // == Fraction(5, 1)

cout << "Result of fract7 += 2: " << fract7.print() << endl;
cout << "Result of fract8 -= 3: " << fract8.print() << endl;
cout << "Result of fract9 *= 4: " << fract9.print() << endl;
cout << "Result of fract10 /= 5: " << fract10.print() << endl << endl;
```

In-Class Exercise: Application Code Part 3

```
    // Testing binary arithmetic operators
    Fraction fract7(3, 5);

    Fraction fract71 = fract7 + Fraction(2);
    Fraction fract72 = fract7 - Fraction(3);
    Fraction fract73 = fract7 * Fraction(4);
    Fraction fract74 = fract7 / Fraction(5);

    cout << "Result of fract7 + 2: " << fract71.print() << endl;
    cout << "Result of fract7 - 3: " << fract72.print() << endl;
    cout << "Result of fract7 * 4: " << fract73.print() << endl;
    cout << "Result of fract7 / 5: " << fract74.print() << endl << endl;

    // Q2. Demonstrate that your chosen operator works correctly by following these steps:
    // print out “your chosen operator: “.
    // print out the input values.
    // print out the expected output from the manual.
    // print out the output from the operator overloading.
    // print out the comparison result indicating whether they are the same or not.
    // repeat this test at least two more times.

    return 0;
}
```

End of Class

Thank you

E-mail: youngcha@konkuk.ac.kr