CSE225L: Data Structures and Algorithm Lab

Lab 03: Operator Overloading

North South University

Functionalities of operators in C++ are predefined but we can overload their characteristics. This is particularly useful for user defined classes specially if they represent custom datatypes like fraction, 2d points or complex numbers. Even though we can overload a operator to make it work the way we want; for example: making subtraction operator (-) to add two complex numbers, it is recommended to not do so.

In the following example we have overloaded multiplication operator (*) to multiply two fractional numbers.

Fraction.h

```
#ifndef FRACTION_H_INCLUDED
#define FRACTION_H_INCLUDED
#include <iostream>
using namespace std;
class Fraction{

private:
    int numerator;
    int denominator;

public:
    Fraction();
    Fraction(int, int);
    int getNumerator();
    void setNumerator(int);
    int getDenominator();
    void setDenominator(int);
    Fraction multiply(Fraction);
    Fraction operator*(Fraction);
    void print();

};
#include "Fraction.tpp"
#endif // FRACTION_H_INCLUDED
```

Fraction.tpp

```
#include "Fraction.h"

Fraction::Fraction(){

    numerator = 0;
    denominator = 1;

}

Fraction::Fraction(int numerator, int denominator){
    this->numerator = numerator;
    this->denominator = denominator;
}

void Fraction::print(){

    cout<<numerator<<"/"<<denominator<<endl;

}

int Fraction::getNumerator(){
    return numerator;
}
void Fraction::setNumerator(int numerator){
    this->numerator = numerator;
}
int Fraction::getDenominator(){
    return denominator;
}
void Fraction::setDenominator(int denominator){
    this->denominator = denominator;
}
Fraction Fraction::multiply(Fraction f){
    Fraction result;
    result.numerator = numerator*f.numerator;
    result.denominator = denominator*f.denominator;
    return result;
}

Fraction Fraction::operator*(Fraction f){
    Fraction result;
    result.numerator = numerator*f.numerator;
    result.denominator = denominator*f.denominator;
    return result;
}
```

.

Tasks:

1. Add 3 functions in Fraction class to add, divide and subtract fractions using overloading respective operators.

2. In the driver file, create two Fraction objects f1 and f2. First one should be created using no argument constructor and second one with arguments 3 and 5.
    a. Set numerator and denominator of f1 to 2 and 6 respectively.
    b. Now add, multiply, divide and subtract between f1 and f2 and print the results using print function in Fraction class.

3. Create a class that represents time. Now overload addition operator (+) so that summation of two time objects can be obtained.

   In the driver file create two time objects and add them. Output the result using print function.

   N.B: Time class should have variables to store hour, time and second and print function to print time in **hour:minute:second** format.