COSC 501, Lab 11

Program 1: Class

Write a C++ program that conducts the rational number operations. Your program will use a rational number class to perform the 'addition', 'subtraction', 'multiplication', and 'division' operations on two fractions.

Use constructors for correctly initializing variable members, and creating objects with given initial values.

Note that a fraction may have both a numerator and a denominator, for example: **(8/3)**, or may simply be a number, for example: **6**, in which case you will actually use **(6/1)** to represent it. Thus, the constructors with two arguments and one argument should be included in the class definition.

Suppose we have defined rational number class objects a and b as two fractions,
then to compute **a + b**, we will use: **a.add(b)**.

When you conduct an operation, you do not have to simplify the result, i.e., **4/5 * 5/10 = 20/50**.

Here are some of the rules you may need:

```
(a/b) + (c/d) = (a*d + b*c) / (b*d)
(a/b) – (c/d) = (a*d - b*c) / (b*d)
(a/b) * (c/d) = (a*c) / (b*d)
(a/b) / (c/d) = (a*d) / (c*b)
```

You will use this code to test your program:

```
int main() {
    RationalNumber rn1(1,6);
    RationalNumber rn2(2);

    cout << "First number: ";
    rn1.printNumber();
    cout << "\nSecond number: ";
    rn2.printNumber();

    cout << "\nAddition: ";
    rn1.add(rn2);
    cout << "\nSubtraction: ";
    rn1.subtract(rn2);

    cout << "\nMultiplication: ";
    rn1.multiply(rn2);
    cout << "\nDivision: ";
    rn1.divide(rn2);
    return 0;
}
```

Sample Output: Red colored texts are user inputs. Other texts are the output of the program.

```
First number: 1/6
Second number: 2/1
Addition: 13/6
Subtraction: -11/6
Multiplication: 2/6
Division: 1/12
```

Program 2: Class Inheritance

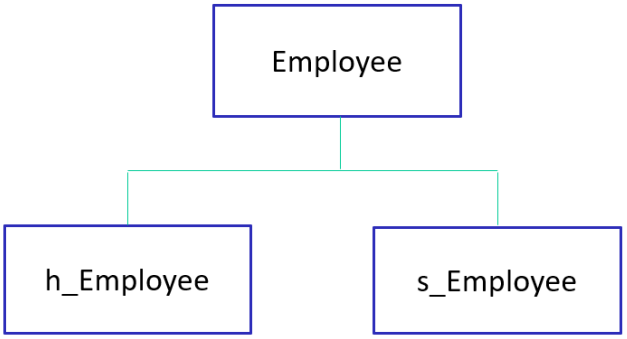
A class **Employee** is a base class of its derived classes **h_Employee** (hourly) and **s_Employee** (salaried).

Employee has member variables (**name** and **netID**) and functions:
constructor, get_name, get_netid, change_name, change_netid, and **print_info**.

Employee can take 2 parameters or none as initial values.

The get functions will return the value, and the change functions will replace the value with new one.
The **print_info** function will print employee's name and netID.

h_Employee and **s_Employee** have only constructors, and other member variables and functions are inherited from **Employee**.



Sample Output:

```
cslabadm05$ .\_employee.exe

        #Joe Smith, employee...
        #employee constructor...
                emp_name=smith, emp_netid=1864
                emp Name: smith
                emp netID: 1864
                emp_name=smith, emp_netid=1865

        #William Ford, employee...
        #employee constructor...
                emp_name=ford, emp_netid=1900
                emp_name=ford, emp_netid=1900

        #Helen Drew, h_employee...
                hourly Name: drew
                hourly netID: 2020

        #Init values, h_employee...
                emp_name=jones, emp_netid=0000

        #Kat Norris, s_employee...
                emp_name=norris, emp_netid=3210
```