CS 7A Final Exam - Spring 2015 - Final Exam 5/19/15

Write responses on separate paper.

1. Define each of the following terms in the context of this course. Give an example of how each may be used.

   (a) *conditional statement*

   (b) *for statement*

   (c) *while statement*

   (d) *operator precedence*

   (e) *type*

2. Consider the program shown below:

```
int main() {
    ofstream ofs("data.txt");
    cout << "\nEnter␣four␣positive␣integers␣followed␣by␣ctrl+z:␣";
    int x{0};
    while(cin>>x) ofs<<x<<"␣";
    ifstream ifs("data.txt");
    int a{0};
    ofs.close();
    while(!ifs.eof()) {
        ifs>>x>>a;
        while (x >= 2) {
            cout << a << "␣";
            x = x - 1;
            if (x % 2 == 0) {
                a = a + x;
            }
            else {
                a = a - x;
            }
        }
        cout << endl;
    }
}
```

Suppose the user responds to the prompt like so:

```
     Enter four positive integers followed by ctrl+z:  5 0 7 0 ∧Z
```

(a) What will become of the file `data.txt` ?

(b) Show the list of values taken by the variable `a` that are output to the console as the program executes.

3. It is known that $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$. The program below uses this to approximate $\sin(x)$ for user-supplied values of x:

```
 1 double sine(double x) {
       const double twoPi = 6.28318530718;
 3     double sum{0};
       double tolerance{ 1e-3};
 5     double n{0};
       while(abs(x)>twoPi) x -= twoPi;
 7     double next = x;
       do {
 9         sum += next;
           n += 2;
11         next *= -x*x/(n*(n+1));
           cout << "\nnext␣=␣" << next << ",␣sum␣=␣" << sum;
13     } while(abs(next)>tolerance);
       return sum;
15 }

17 int main() {
       double x;
19     cout << "\nEnter␣a␣value␣for␣x␣to␣compute␣sin(x):␣";
       while(cin>>x) {
21         cout << "\nsin(" << x << ")␣=␣" << sine(x);
       }
23 }
```

(a) Provide a definition for `abs()` to take a double and return its absolute value.

(b) What does the while loop on line 6 do? Why?

(c) The while loop on on line 6 will not behave properly for some negative x. What's wrong? How would you fix it?

(d) The series formula for $\cos(x)$ is $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$. Modify the `sine()` function to produce a `cosine()` function.

(a) which parameters are passed to `score_play()` by reference. Why is that needed?

(b) Specify the appropriate parameters to pass to `score_play()` in `main()` on line 21.

(c) Write the additional code to make `score_play()` work correctly (code to be inserted at line 11.)

4. Help complete the design and implement a rational number class, `Rational` started below by answering the questions that follow. Addition, and multiplication operator overloading are provided.

```
int gcd(int x, int y) {
        // greatest common divisor using Euclid's algorithm
        x = abs(x);     // don't get confused by negative values
        y = abs(y);
        while (y) {
                int t = y;
                y = x%y;
                x = t;
        }
        return x;
}
//--- the class -----------------------------------------------------------------
class Rational {
public:
        Rational(int n, int d) :num(n), den(d) { normalize(); }
        Rational() :num(0), den(1) { }

        void normalize() {// keep denominator positive and minimal
                if (den==0) error("negative␣denominator");
                if (den<0) { den = -den; num = -num; }
                int n = gcd(num,den);
                if (n>1) { num/=n; den/=n; }
        }
        int num, den;
};
Rational operator+(Rational x1, Rational x2)
{
        Rational r(x1.num*x2.den+x1.den*x2.num, x1.den*x2.den);
        r.normalize();
        return r;
}
Rational operator*(Rational x1, Rational x2)
{
        Rational r(x1.num*x2.num,x1.den*x2.den);
        r.normalize();
        return r;
}
ostream& operator<<(ostream& os, Rational x) {
        return cout << '(' << x.num << '/' << x.den << ')';
}
```

(a) Explain what the `gcd()` function does and how it is used here.

(b) Describe the constructor function for the `Rational` class. What parameters are passed? What is the initializer list? What happens in the body of the function? What would the statement `Rational r2(40,24);` do?

(c) Write functions to overload the subtraction(-) and division(/) operators.