

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:

```
1 int main() {
    ofstream ofs("data.txt");
3  cout << "\nEnter four positive integers followed by ctrl+z: ";
    int x{0};
5  while(cin>>x) ofs<<x<<" ";
    ifstream ifs("data.txt");
7  int a{0};
    ofs.close();
9  while(!ifs.eof()) {
        ifs>>x>>a;
11     while (x >= 2) {
            cout << a << " ";
13         x = x - 1;
            if (x % 2 == 0) {
15                 a = a + x;
            }
17         else {
                a = a - x;
19         }
        }
21     cout << endl;
23 }
```

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 $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$. 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 }

```

- Provide a definition for `abs()` to take a double and return its absolute value.
- What does the while loop on line 6 do? Why?
- The while loop on on line 6 will not behave properly for some negative x . What's wrong? How would you fix it?
- The series formula for $\cos(x)$ is $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$. Modify the `sine()` function to produce a `cosine()` function.
 - which parameters are passed to `score_play()` by reference. Why is that needed?
 - Specify the appropriate parameters to pass to `score_play()` in `main()` on line 21.
 - 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.

```

1  int gcd(int x, int y) {
    // greatest common divisor using Euclid's algorithm
3   x = abs(x);    // don't get confused by negative values
    y = abs(y);
5   while (y) {
        int t = y;
7       y = x%y;
        x = t;
9   }
    return x;
11 }
//--- the class -----
13 class Rational {
public:
15     Rational(int n, int d) :num(n), den(d) { normalize(); }
    Rational() :num(0), den(1) { }
17
    void normalize() { // keep denominator positive and minimal
19         if (den==0) error("negative denominator");
        if (den<0) { den = -den; num = -num; }
21         int n = gcd(num,den);
        if (n>1) { num/=n; den/=n; }
23     }
    int num, den;
25 };
Rational operator+(Rational x1, Rational x2)
27 {
    Rational r(x1.num*x2.den+x1.den*x2.num, x1.den*x2.den);
29     r.normalize();
    return r;
31 }
Rational operator*(Rational x1, Rational x2)
33 {
    Rational r(x1.num*x2.num,x1.den*x2.den);
35     r.normalize();
    return r;
37 }
ostream& operator<<(ostream& os, Rational x) {
39     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.