CECS 282 - Homework 7

Complete these problems on a separate sheet of paper. Due March 17.

1. Reading from *C++ How to Program*:
   (a) Chapter 10.9 **again**
   (b) Chapter 10.12, 10.15
   (c) Chapter 15.1, 15.2

2. Consider the following short main function, using your `Rational` class from Lab 4/5:

```
int main() {
   Rational a(5, 4);
   if (true) {
      Rational b(2, 1);
      Rational *c = new Rational(4, 5);
   }
}
```

   (a) How many `Rational` objects are constructed in this example?
   (b) How many `Rational` objects are destructed by the time `main` ends, but before the operating system closes the program?
   (c) Is `c` a `Rational` object? Why or why not?

3. Answer True or False to the following questions:
   (a) You can declare a pointer to point to a value on the heap.
   (b) You can declare a pointer to point to a value on the stack.
   (c) It is safe to delete a pointer when it points to a value on the heap.
   (d) It is safe to delete a pointer when it points to a value on the stack.
   (e) A function can determine at run-time whether a pointer points to a stack or heap value.
   (f) It is safe to blindly delete any pointer.

4. In your own words, describe the differences between the three places "const" can appear in a function prototype. You can refer to the following prototype as an example:
   `const Pokemon& Pokemon::DoSomething(const Pokemon &parameter) const;`

5. Consider the following code fragment using your `Rational` class from Lab 4/5:

```
const Rational r2(3, 4);
r2.SetNumerator(10);
cout << r2.ToString();
cout << r2.GetNumerator();
```

   (a) Which of the three function calls on lines 2-4 is **not allowed** by the C++ compiler? Why?
   (b) How does the compiler know that your answer to (a) isn't allowed? **Hint**: the compiler **does not** look at the **implementation** of the functions.

6. Read about the C++ standard library classes `std::istringstream` and `std::ostringstream`. Answer the following questions:
   (a) What library header must you `#include` to use these classes?
   (b) Suppose you have a variable `std::string s = "100 200 300";` Show how to use a `std::istringstream` to "read" the three integer values in the string into three `int` variables. (This is called **parsing**.)
   (c) Re-implement the `std::string Rational::ToString() const` method from Lab 4 to use an `std::ostringstream` object to construct the return value, **instead of** `std::to_string` and `operator+`.