

CS250 Exam#2

Review Topics

- 1) What is the difference between procedural and object-oriented programming?
- 2) What is the difference between a class and a class instance?
- 3) Why not just define all class members to be public?
- 4) Understand composition from a theoretical and implementation level.
- 5) Remember, class declarations (interfaces) are stored in a class specification file (.h) while member function definitions are stored in a class implementation file (.cpp). Why do we separate out the interface in a .h file and the implementation in a .cpp file?
- 6) What is an object request? Give an example. How have we dealt with object requests?
- 7) When are constructors called?
- 8) What is a default constructor? How can a default constructor be defined?
- 9) What is overloading? What can be overloaded?
- 10) Review access rights of static versus non-static member functions on static and non-static data
- 11) If class X declares function f as a friend, does function f become a member of class X.
- 12) Consider the following class declaration and program segment. Assuming z has been properly initialized, answer each question below.

```
class Foo
{
    private:
        int x;
        int y;
        static int z;
    public:
        Foo ()
        {
```

```

        x = y = z;
    }
    static void set (int value)
    {
        z = value;
    }
};

```

```

// program segment
Foo cFool, cFoo2[5];
cout << "Spring Break" << endl;

```

- a) How many separate instances of the x member exist right before Spring Break is printed out?
- b) How many separate instances of the z member exist right before Spring Break is printed out?
- c) How many times is the constructor called during the execution of the above program segment?
- d) How would you initialize z to a value?

13) Make sure you finish class Rational and RationalSet and that you understand every statement in the implementation of Rational and RationalSet. You will most likely see the concepts on the exam in one form or another. You need to be able to write code without your notes.

14) What is a friend?

15) Operator overloading.

16) Look over problems specified in the lecture notes. There are many good test type questions.