

CS 250 Exam 2 Review

Vocabulary: class, object, interface (.h), implementation (.cpp), constructor, mutator, accessor, scope resolution operator, encapsulation, ADT, private, public, data members, member functions, default constructor, default arguments, primitive data types, UML, operator overloading, static, pointers (dereference, indirection, address of), dynamic memory (new, delete), this, memory leak, null pointer, pointers and functions, const pointers, pointers to const

1. How is a static data member different than a regular data member in a class?
2. How is a friend function different from a member function?
3. Why does operator>> need to be a friend function?
4. What is overloading? What can be overloaded?
5. If class X declares function f as a friend, does function f become a member of class X.
6. 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 cFoo1, 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?
7. List three uses of * in C++
 8. Assume plnt points to a 4-byte int and has the address 500. What is the value of plnt after: a) ++plnt; b) plnt += 5;

9. A double array **aValues** contains **size** values in the array. The median value is the middle value. If the array contains an even number of values, the median is the average of the two middle values. Write a function **median** that accepts the array **aValues** and **size** and returns a double representing the median of the array of values. Use only pointer notation.