CS 330 Exam #3 Study Guide
Chapters 8, 10-12

The questions at the back of the chapters are also an excellent source for studying for this exam.

1. What are the two mechanisms by which the order of evaluation of expressions is controlled by a programming language? Describe them.

2. What are the four kinds of sequence control provided by most programming languages?

3. There are four basic models for repetition. Why? Describe them (give an example).

4. Describe the “dangling else” issue and give an example. What are two ways this issue can be resolved?

5. What are the primary design issues for multiple selection statements?

6. Describe the contents of a typical activation record.

7. What are the differences between an activation stack’s static and dynamic links? When does a programming language need a static link in its activation record?

8. When local referencing environments are deleted between subprogram activations, using a central stack as in C/C++, it sometimes appears as if values are retained. For example, if procedure Sub has a local variable X and Sub assigns the value 5 to X on the first call, then on a second call, if X is (inadvertently) referenced before it is assigned a new value, sometimes X still has its old value 5. However, in the same program, a third call on Sub may find X has not retained its old value from the second call. Explain this apparent anomaly. In what circumstances could an activation that references an uninitialized variable find that that variable still had a value assigned on a previous call? In what circumstances would it not have its previously assigned value (but may have some other value entirely unrelated to that variable)?

9. Give an example in code (either Java or C++) of causing the program stack to run out of space.

10. Give an example in code (either Java or C++) of causing the program heap to run out of space.

11. Given a program in a language that allows recursion and uses static scope, show the activation record at a given point in the program’s execution.

12. Given a program in a language that allows recursion and uses dynamic scope, show the activation record at a given point in the program’s execution.
13. Given a program in a language that allows nested subprograms, show the activation record at a given point in the program’s execution.

14. What are the two most important principles in the construction of abstract data types? Describe each.

15. What are the differences between the support for abstract data types in C++ and Java?

16. What are the information hiding mechanisms of Java? How are they different from similar ones in C++?

17. What are the fundamental design issues for object-oriented languages?

18. Give an example of the inheritance mechanism in C++ and Java.

19. Given the following pseudo-code in some arbitrary language:

```plaintext
Class Aclass {
    method print...
}
Class Bclass is_a A {
    method print...
}
Aclass X;
Bclass Z;
... X.print;
... X = Z;
X.print;
```

What are the concerns about the second X.print operation? Why?

20. Consider the following C++ code:

```cpp
class single_linked_list {
private:
    class node {
        public:
            node *link;
            int contents;
    };
    node *head;
public:
    single_linked_list () {head = 0};
    void insert_at_head (int);
};
```
```cpp
class stack : public single_linked_list {
public:
    stack ( ) { }
    void push (int value) {
        insert_at_head(value);
    }
    int pop ( ) {
        return remove_at_head( );
    }
};
```

What are the issues involved in how the stack object is dealt with by C++? How is the stack class’s constructor handled? What about the stack class’s need for an empty( ) method? Are there any hazards with this implementation of a stack ADT?