Name ______________________

Section ______________

CS2303 C14
Systems Programming Concepts
Final Exam
March 7, 2014

Question	Points	Score
0	1	
1	8	
2	8	
3	6	
4	3	
5	2	
6	8	
7	4	
8	5	
9	6	
10	4	
11	5	
12	5	
13	6	
14	4	
15	6	
16	10	
17	10	
Total	100	

Trivia Question (1 extra credit point)

0. (a) What is the capital of Cambodia?

-OR-

(b) Who won best supporting actress at the Oscars last Sunday?

(4 pts) 1a. Starting from the following input: M, F, G, Y, R, J, B, A, use the **insertion algorithm** for a **binary tree** to build the binary tree that keeps the elements of the tree in **alphabetical order**. Draw the final tree.

(4 pts) 1b. What is the output of an **inorder traversal** of this tree that prints each node's contents when it visits the node?

(3 pts) 2a. Explain the term **concurrency** in the context of an event-driven simulator or a cpu scheduler.

(5 pts) 2b. Define the difference between **transmission time** and **propagation time** when a packet is sent over a computer network.

(6 pts) 3. What are the desirable properties of a good **hash function**?

(3 pts) 4. How is a **linear list** different from a **linked list**?

(2 pts) 5. Explain how the syntax changes in a called C++ function that provides for **pass by reference**.

(8 pts) 6. Explain the difference between using **default arguments** and **member initializers** in a C++ **object constructor**. Provide an example (that is a snippet of code) that shows clearly the syntax differences for these two concepts for an object constructor where the object includes two private integer data members.

(4 pts) 7. Why are **set** and **get** functions used in C++?

(5 pts) 8. What determines the **sizeof** a C++ **object**?

(6 pts) 9. Name three techniques for referencing a C++ public class member from outside a class's scope.

(4 pts) 10. Explain how and why **#ifndef** is used in C++ header files.

(5 pts) 11. Explain what happens when a C++ **destructor** is called.

(5 pts) 12. When is **member initialization** required in C++?

(6 pts) 13. What is a **default copy constructor**? How is it used in **object composition** in C++?

(4 pts) 14. When a program **overloads a function**, how does the C++ compiler decide which of the overloaded functions to execute?

(6 pts) 15. How does a **static** data member differ from a **non-static** data member for a C++ class?

(6pts) 16a. What are the two possible approaches for implementing **operator overload** in C++?

(4 pts) 16b. Name two C++ **operators** that **cannot** be overloaded.

(6 pts) 17a. How does a class **derived** from a **public base class** access **private members** of the **base class**?

(4 pts) 17b. List four things **not inherited** by **derived classes**.