Jeremy Scheuerman

Final Exam Review

COSC 220

Dr. Wang-

@all 1) Concepts to be focused on: OO programming concepts (inheritance, polymorphism, static/dynamic binding, virtual function, abstract class).

**OO Programming**

(one small question on inheritance/poly morphism)

**Inheritance**-is when a child class inherits data members of a parent class.

**Polymorphism-**happens when the program determines what function to call during runtime instead of

determining what function to call when the program is compiled.

**Bindings**

**Static**-program knows exactly what functions to call at compile time

**Dynamic-**the program doesn’t know what functions to call at compile time and must determine which ones to call during run time. (Binding is unknown at compile time and is determined during run time)

**Virtual Function-**a member function that is declared in a class and is overridden by a derived class, used to achieve runtime polymorphism

**Abstract Class—**a class that has at least one virtual function, essentially a base class with a virtual function

2) Programming components focused on:

2.1) use of pointers for dynamic array creation and when a pointer is a data member of a class implementation of constructor/destructor/copy constructor/assignment operator overloading (look at how the mini-vector and the lab of week 13) are implemented

2.2) STL library we discussed,

**Vector-**similar to an array, can be expanded and copied

**stack** (adds to the top)-adds new element to the top

**queue** (adds to the bottom/back)

-uses front()\_, top()

**list -**controlled with an iterator, has .begin , has push back and push front members, so you can add data to the front and the back of the list, along with pop\_back and pop\_front, even has options to reverse the list and stuff\

3) Performance analysis components to be focused on:

3.1) perform analysis of simple instructions such as i = i+1, and a for loop using a detailed model (with Ts) and 3.2) perform analysis of familiar algorithms such as finding the largest number in an array, display a 2D array using a simplified mode (without Ts, just a function of n, which is the size of an array)

3.3) know the big-O all the search (linear, binary) and sorting (bubble, selection, merge, radix) algorithms. There will be NO prove of big-O of an algorithm (looking for C and n0)

Performance, a performance analysis on a question

4) **Binary Search Tree** 15 to 20% of the questions will be on BST - mainly using recursive function to do some simple operations on a given BST such as count leaves.

-(know how to manipulate code of binary search tree)