**CIS 252 – Exam3 – Part 2 – Advanced C++ Programming**

**Instructions section**

**This section is worth 4 points on your grade.**

* Save a private copy of this file (“Save As” function) in your personal directory or flash drive as a Word file with the filename

**CIS\_252\_T3\_firstname\_lastname.doc(x)**

* Click on the Insert tab in MS Word, find the “Header and Footer” group and click on the Footer button. Select the first footer option, which will put you in the footer. Type the filename in the footer, then double-click back into the text.
* You can type in the answers now and then print or print your private copy and fill in the answers by hand.

**Short Answer section (3 points each).**

This section contains eleven questions.

* You are required to answer any ten of your choosing.
* You may answer the remaining one as consideration for possible bonus points between 1 and 3 points.
  + However, you must indicate which is the extra 1 answered by putting the word **BONUS** to the left of the answer to be considered for any bonus.
  + There may be points given for it.

1. Explain the difference between a shallow copy and a deep copy of data.

A shallow copy is two pointers that use the same memory location, and a deep copy is two pointers that have their own memory but the same data\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. If we are going to use a vector object in our program, which header file would need to be included in the program?

#include<vector>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

For 3 – 5 Please list three functions that can be used to perform operations on a vector and the effect each one has on a vector object.

1. vectorName.at(index) returns the value of position index \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. vectorName.push\_back(type) adds a copy of variable type to the end of the vector list \_\_\_\_\_\_\_\_\_\_\_\_
3. vectorName.clear() deletes all values in the vector \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What aspect of a recursive function can cause issues with memory?  
     
   **BONUS** if the recursive function never gets to a base case\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Give two characteristics/requirements of a Binary Search and describe how the Binary Search works.  
      
   the list must be sorted the item being searched is compared to the middle of element of the list to determine whether to search the upper or lower half of the list and this process is repeated until the search is complete
3. What is a tail recursive function?  
     
   a recursive function in which the last statement of the function calls itself \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. What is a dynamic variable? Describe when and how it is created and what happens when it is no longer used.

A dynamic variable is a variable created during program execution using the new command and is removed using the delete command once it is no longer needed \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Compare and contrast a static array versus a dynamic array.

A static array remains unchanged and is created during compile and a dynamic array is user defined and created during program execution \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Consider the code below:

int \*num;

num = new int[25];

\*num = 325;

What would each of these statements do?

int \*num; creates a pointer variable of type integer named num \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

num = new int[25]; allocates memory for a dynamic array of 25 elements to which num is pointing

\*num = 325; stores the value of 325 into the first memory space of num \_\_\_\_\_\_\_\_\_\_

**Code comprehension** (18 points).

Determine whether the following code fragments are 1) missing code, 2) valid code or 3) if the ones given contain errors.

* Where there is missing code, Type **M** and **write in what is missing,**
* Where there is an error, Type **E** to **indicate the error and write the corrected line next to or below it**.
* Where there is no error, Type **N** for **“no error**”.
* There are supposed to be 36 statements of code including braces.
* There are 18 lines of code with no errors
* There are 18 lines of code with either errors or are missing
* FAILURE TO MARK LINES AND PUT IN CORRECTED CODE WILL COUNT OFF!

1. The first line below is the data that is used in the program.

//Data: John Bill Danny Minnie Lisa Jack Susan Pluto zzz

\_N\_1 #include <iostream>

\_E\_2 #include <string>; #include<string>

\_E\_3 include <vector> #include<vector>

\_N\_4 using namespace std;

\_N\_5 int seqSearch(vector<string> list, string searchItem);

\_M\_6 int main()

\_N\_7 {

\_N\_8 vector<string> nameList;

\_E\_9 name string; string name;

\_N\_10 int index;

\_E\_11 cout << "Enter first names (zzz to end the input)." << endl;

cout << “Enter first names (zzz to end the input).”;

\_N\_12 cin >> name;

\_E\_13 while (name != "aaa") while(name != ‘zzz’)

\_N\_14 {

\_E\_15 namePush.list\_back(name); nameList.push\_back(name);

\_N\_16 cin >> name;

\_M\_17 }

\_N\_18 cout << endl;

\_N\_19 cout << "Enter the name to be searched: ";

\_E\_20 cin << names; cin >> name;

\_N\_21 cout << endl;

\_N\_22 index = seqFind(nameList, name);

\_N\_23 if (index = -1)

\_E\_24 count << name << is found in the list." << endl;

cout << name << “ is not found in the list.” << endl;

\_M\_25 else

\_E\_26 cout << name << " is in the list." << endl;

\_M\_27 return 0;

\_N\_28 }

\_E\_29 int seqSearch(vector<string> list, string searchOne)

int seqSearch(vector<string> list, string searchItem)

\_N\_30{

\_E\_31 int len = static\_cast<int>(list.size();

int len = static\_cast<int>(list.size());

\_E\_32 for (loc = 0; loc < len; loc++);

for(loc = 0; loc<len; loc++)

\_E\_33 { if (list[loc] = searchItem)

\_E\_34 return location; }

\_N\_35 return -1;

\_N\_36 }

**BONUS: Programming problem** (12 points)

For this portion of the exam, you will need to create a program to test the insertion algorithm. This is in chapter 10.

You will need to use a length of 20 for the list.

You will need to assign 20 random numbers that are not in order to the values of the list.

You can use the function in the book, but you will need to code the rest (main). Make sure that you call the function and are using a for statement for output.

This will be contained in one .cpp file.

Make sure to have an initial documentation section in the program.

Put description comments in important parts of your code.

When you are finished and have a successful program (that both compiles and runs), submit the .cpp file to this assignment. File name should be,

**CIS252\_T3\_prog\_<first name\_last name>.cpp**