CSCI 2170 Final Exam **Name _____**

You are on the honor system and you are to do this exam with no help from any other person and the internet. When you sign your name below, you are indicating that you have adhered to these restrictions.

Email me (Cen.Li@mtsu.edu) in case you have questions.

1. For the **Final exam paper test component (70 pts)**
   - include the following line at the top of your answer sheet, with your name filled in.

   I, _____, worked on all of the problems on this test completely on my own without any assistance from any other person or the internet.

   - Just type the answers, label each answer with the question number.
   - Save the file and export it as a **PDF file** before submitting to the D2L Dropbox labelled "Final Exam".
   - The paper test is due midnight Wednesday May 6th . Make sure to submit the answer to D2L dropbox before due time.

2. For the **Final exam programming component(30 pts)**, you need to write, compile and run the program on the "ranger" system and submit the program using the "handin" command:

   handin final car.h car.cpp main.cpp

   If you work on the bonus program: handin finalBonus stack.cpp

   The programming test is due midnight Wednesday May 6th. Make sure to handin the program before the due time.

Final Exam Paper Test Questions:

1. (10 pts) Short answer questions:

   a. When two objects, obj1 and obj2, of a user defined class are compared using: obj1 == obj2, the operator == is said to be _____.

   b. The main characteristic of Queue ADT is _____, and the main characteristic of the Stack ADT is _____.

   c. The value of a pointer variable is _____

   d. A _____ is a function which is called automatically when an object of the class is created.

   e. In C++, the **delete** operation is used to _____

   f. A _____ is an appropriate ADT to use to simulate the customers waiting to buy the movie tickets.

   g. In C++, the _____ directive is used in the header file to prevent Multiple Inclusion.

   h. _____ is used to create an alias of an existing or user defined type.

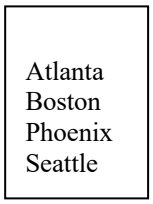
   i. A destructor needs to be defined in a C++ class when _____ _____

2. (3 points) Evaluate the postfix expression below. Show the final result. Show intermediate results for partial credits.

   `1 4 3 - / 5 – 4 2 * +`

1 (10 points) Assuming a **stack** object named **"myStack"** contains a sequence of cities represent the path from an origin city to a destination city, with the origin city at the bottom of the stack and the destination city at the top of the stack. You are required to use the **C++ STL stack container** for these two problems.

Write the C++ function **PrintPath** that prints the itinerary of the path in the form of a sequence of "From *city* to *city*" pairs. For example, with "myStack" shown on the previous page, the output of your program should be the following



← bottom of the stack

myStack

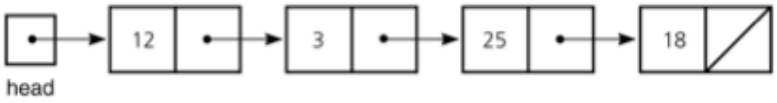
Here is the itinerary:
From Seattle to Phoenix
From Phoenix to Boston
From Boston to Atlanta

void PrintPath(stack <string> & myStack) { // this is the function heading

2 (6 pts) Show the output of the following program?

```
const int SIZE = 20;   // maximum number of items to store in array
void MyFunc(int arr[], int pos, int &size);
int main(){
        int array[SIZE];
        int position=2;
        int aSize=10;

        for (int i=0; i<aSize; i++)
            array[i] = i*2;

        MyFunc(array,  position, aSize);

        for (int i=0; i<aSize; i++)
            cout << array[i] << " ";

        return 0;
}
void MyFunc(int arr[], int pos, int &size){
        for (int i=pos; i<size-1; i++)  {
            arr[i] = arr[i+1];
        }
        size--;
}
```

3 (6 pts) For this question, only the head of the **linked list** is given.



Show C++ loop statements needed to compute the average of all the values stored in the linked list.

4 (20 pts) For this problem, you are required to use the unsorted list class array implementation as discussed in class. Write a C++ **client program** to read in a number of integer values from a data file and store the numbers in a list. The values are stored one value per line in the data file. After all the values are inserted into the list, call user-defined function "**AboveAndBelow**" to display all the values that are above the average value, and then to display all the values that are below the average.

```
// Assume all the needed header files have already been included
// Declare the user-defined functions here

int main()  {
    int count=0, data;
    ifstream myIn("data");  assert(myIn);
    float average;
    // declare a list class object here

    // read values from the data file and add to the list, repeat til the end of the data file is reached

    // call "ComputeAverage" function compute the average value of the list of numbers.

    // call "AboveAndBelow" function to display the values in the list that are above the average, and
    // to THEN display the values in the list that are below the average

    return 0;
}

// define the function "ComputeAverage" here

// define the function "AboveAndBelow" here
```

5 (15 pts) Show a C++ value returning function that takes two sorted linked list and merge them into one single sorted linked list. The parameters of the function are the head pointers of the two sorted linked list. The function return value is the head of the newly merged linked list.