Name:			

## CS 161 Week 7 Worksheet:

## **Pointers and Arrays**

## **Pointers:**

- 1. What is one difference between pointers and references?
- 2. Predict the output of this code:

```
short* leaf = new short;
*leaf = -10;
cout << *leaf << endl;</pre>
short* tree = new short[4];
tree[0] = *leaf;
tree[1] = *leaf + 1;
tree[2] = tree[0] * 20;
tree[3] = 15 - tree[1];
for (int i=0; i<4; i++) {
   cout << tree[i] << " ";</pre>
cout << endl;</pre>
*tree += 2;
short* rock = &(tree[2]);
cout << *rock << endl;</pre>
rock++;
cout << *rock << endl;</pre>
```

- 3. Write a C++ statement that uses rock (without changing rock) to change tree[1] to 15. (Think creatively!)
- 4. What delete statements should come after the above code segment to clean up the heap and avoid memory leaks?
- 5. What happens if you delete tree; instead of delete [] tree; ?

## 1-D and Multidimensional Arrays:

1. What error was made in this program? How would you correct it?

```
int* get_ducks() {
    int duck[10] = {};
    duck[3] = 47;
    return duck;
}

int main() {
    int* my_ducks = get_ducks();
    cout << my_ducks[3];
    int goose = 3;
    cout << my_ducks[3];
    return 0;
}</pre>
```

- 2. Define these terms:
  - a. Multi-dimensional array
  - b. Row-major
  - c. Column-major
  - d. Stride
- 3. Give an example of something from the real world that could be modeled with a multidimensional array.
- 4. Does C++ use row-major or column-major array layout in memory?
- 5. Static vs. dynamic 2D arrays:
  - a. How do you create each kind?

	b. Where are they located and how are they laid out in memory?
	c. How do you pass each kind of array to a function?
6.	Write a void function that will assign every element in a static 3 by 5 integer 2D array to 42. What parameters would we need for this function? How does the function prototype and function call differ between statically versus dynamically allocated arrays?
7.	Write a function to allocate memory for a 2D array in a function, given any number of rows and columns. Show an example of how to call your function with your favorite number of rows and columns. Show how to clean up afterwards to avoid a memory leak.