

Lab 3: Dynamic Memory

Due: Friday 2/10 at 11:59 PM

(You can type your answers on the [answer sheet](#) so that it can be submitted electronically.)

Part One: (Static and automatic variables.)

1. If you have not already done so, make a directory for your CS2401 stuff
`mkdir cs2401`
2. Go into that directory
`cd cs2401`
3. Once you are in there make a directory for your labs and inside that one a directory for lab4.
4. Open a new file in your favorite editor, type `#include <iostream>` and put in your using statement.
5. Type in this little function:

```
void stars(){
    int x = 0;

    x++;

    for(int i = 0; i < x; ++i){
        cout << '*';
    }

    cout << endl;
}
```

6. Write a `main` that has a loop that calls your function six times.
7. On your [answer sheet](#) write the output that you see after you have compiled and run this program.
8. Now change the first line in the function `stars` so it looks like this:

```
static int x = 0;
```

9. Recompile and run this program.
10. Write the output for this function on your [answer sheet](#).
11. What was the effect of using static? (Write on [answer sheet](#).)
12. If you remove static what is the effect? (Write on [answer sheet](#).)

Part Two: (Dynamic Variables.)

1. Start a new program – again starting out with the `#include <iostream>` and using namespace `std`; - this time you will only need a `main`
2. Declare a pointer capable of pointing at a double.
3. Make the pointer point at a new double.

4. Print out the address of the new double. (On your answer sheet write how you did this as well as the address that shows up.)
5. Print out the address of that pointer. (On your answer sheet write how you did this as well as the address that shows up.)
6. Now, using the * operator, set the value of your double to 24.01.
7. Write a loop with an integer counter that counts to 10 and in the body of the loop does this:


```
++(*ptr);
cout << *ptr << "is stored at " << ptr << endl;
```
8. On your answer sheet write down the first and last lines that appear here.
9. Now change the body of your loop to this:

```
{
    ++(ptr); // notice I took out the *
    cout << *ptr << "is stored at " << ptr << endl;
}
```

The difference here is that you were moving the pointer instead of changing the value being pointed at.

10. On your answer sheet write down what the last two lines of your output looks like.

Part Three (A dynamic array)

1. Now let's make a dynamic array with the pointer that you have been using.
2. Begin by declaring a couple of size_t variables for capacity and used.
3. Initialize capacity to five and used to zero.
4. Declare an extra double* tmpptr, which we will use in resizing.
5. Type this code:

```
ptr = new int[capacity];
for(size_t i = 0; i < 25; ++i){
    ptr[used] = rand(); // you will need #include <cstdlib>
    cout << ptr[used];
    used++;
    if(used == capacity){
        cout << "Sorry no room left.\n";
        break;
    }
}
```

6. Note how many numbers you see in the output.
7. Now replace the cout and the break for a full array with a resizing which will do this:
 - a. Set the tmpptr to a new integer array of capacity +5 size

- b. Copy the data from the original array to the new one using a loop or the copy function which we talked about in class
 - c. Adjust the capacity variable so it accurately reflects the size of the new array
 - d. Delete the original array, remembering to use the []'s
 - e. Assign ptr to tmpptr, so these pointers will both point at the newer array
 - f. Do a `cout << "Resized\n";` to report that this was done.
8. Now run your program – what do you see?
9. Finally, comment out the highlighted code "`cout << ptr[used];`" in step 5, and then at the end of the loop that has filled the array, put this:

```
tmpptr[2] = 0;
```

10. Write a loop that will output all the numbers in the ptr array.
11. What do you see?

Submit the finished code for each of the three parts and your answer sheet to Blackboard, making sure that your name appears on both your code and the answer sheet.