CSCA48 Tutorial 2

Tabeeb Yeamin

tabeeb.yeamin@mail.utoronto.ca

github.com/tabeebyeamin/CSCA48W20

CSEC

PRESENTS

MOCK MY INTERVIEW

Land your dream job.

Friday Jan. 24| 4:00 - 6:00 PM| HL B101

FEATURING INTERNS FROM:

INDEX EXCHANGE Google salesforce shopify

amazon Microsoft confluent TD

Topics Covered Last Week

Unit 2 Notes up to pg ~25

- Memory Model in C - Locker Room with Boxes
    - Uniquely numbered
    - Stored in increasing order
    - Access controlled (only right user can store stuff or see what’s there)
- How information is passed into functions (copies of variable values are made)
    - Local variables, input parameters, return variables
- Arrays
    - Fixed Size, contiguous boxes of the same data type
    - Strings (arrays of characters)
        - ‘\0’
        - How strings are passed into functions (they don’t make copies)

Memory Diagrams

```
int main()
{
    int x;
    int y;
    x=5;
    y=x;
}
```

Draw a memory diagram of the following code.

If you were to modify y, would it modify x?

Arrays

You have to specify a size, they are fixed to that size.

```
int a[3] = {1, 2, 3};
```

Strings are “array of characters”

```
char a[1024] = “hello\0”;
```

What is the output for this code?

```
int main()
{
    char original[1024] = “This is the original string!”;
    char unoriginal[1024] = “And this is another string!”;

    original = unoriginal;

    printf(“%s\n”, original);

}
```

What is the output for this code?

```
int main()
{
     char original[1024] = “This is the original string!” ;
     char unoriginal[1024] = “And this is another string!” ;
     original = unoriginal;
     printf(“%s\n”, original);
}
```

This code will NOT compile. Specifically at the offending line.
Why does it do this?

Assignment of Arrays

```
int main()
{
        int array_one[10];
        int array_two[5];

        for (int i = 0; i<5; i++)
        {
                array_two[i];
        }
        array_one = array_two; // what should happen here?
}
```

Assignment of Arrays

```
int main()
{
        int array_one[10];
        int array_two[5];

        for (int i = 0; i<5; i++)
        {
                array_two[i];
        }
        array_one = array_two; // what should happen here?
}
```

The ‘=’ sign doesn’t have a well defined meaning for arrays.

They can have different sizes.

- Should the compiler copy as many elements as possible and ignore the rest?
- Should it only copy when the arrays are the same size?
- They could have different types:
    - Int array = double array

Write a function that takes two input strings of size 1024 and swaps their content.

The strings could have different length

```
>>> char a[1024] = “hello\0”;
>>> char b[1024] = “today it’s very cold\0”;
>>> swap(a,  b);
>>> printf(a);
today it’s very cold
>>> printf(b);
hello
```