

# CSCA48 - Intro to CS II

TA: Angela Zavaleta-Bernuy

Tutorial: T21 (Mondays BV264 10-11am)

Office hours: 1C402 on Tuesdays 12-1pm

email: angela.zavaletabernuy@mail.utoronto.ca

website: angela.zb.github.io

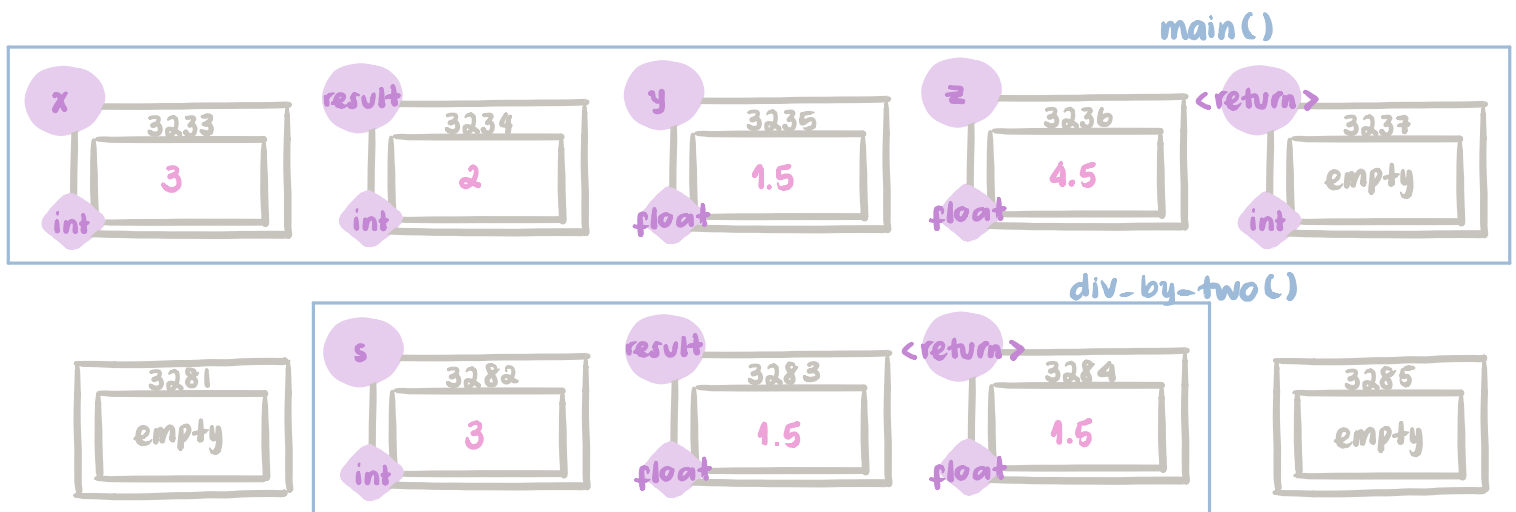
## Week 2 - Jan 20th

### Memory Model in C

- \* It's like a locker room! All locker boxes are numbered in increasing order, and only can be accessed by the right user.
- \* There are 3 different ways in which a program can get a box in memory:
  - Declaring a variable (each variable gets a box)
  - Input parameters to functions (each gets a box)
  - Return value (gets one box)

Ex1: Draw the diagram of the memory model right at the point where the result is returned (before the space reserved for the function is released) for the following program.

```
1  #include<stdio.h>
2
3  float div_by_two(int s){
4      float result;
5      result = s / 2;
6      return result;
7  }
8
9  int main(){
10     int x, result;
11     float y, z;
12     x = 3;
13     y = div_by_two(x);
14     z = y + 3;
15     result = z / 2;
16     printf("The result is: %d\n",result);
17 }
```



# Arrays and Strings

## \* Arrays:

- Collections of contiguous boxes of the same data type. (contiguous in memory)
- Fixed size
- Wrong indexes? You are screwed...

## + Strings:

- Arrays of chars.
- End-of-string delimiter '\0'
- Strings are passed to functions by telling it is location in memory, so the function can modify the original input.

Ex2: What do you think this prints out?

```
1  #include<stdio.h>
2
3  int main(){
4      char original[1024]="This is the original string!";
5      char unoriginal[1024]="And this is another string!";
6
7      original = unoriginal;
8
9      printf("%s\n",original);
10 }
```

does this even work?! why?!

How can we copy elements from an array?

Ex3: Does the following code compile? If not, what would you change?

```
1  #include<stdio.h>
2
3  int main(){
4      int array_one[10];
5      int array_two[5];
6
7      for (int i=0; i<5; i++){
8          array_two[i]=i;
9      }
10
11     array_one=array_two;
12 }
```

Nope! C does not allow assignment between arrays like this.

If we need to copy values we need to do it manually.

Ex4: Write a function that takes two input strings (size 1024) and swaps their content.

Hmm... Interesting!