

Data Structures

CSCI 2270

Algorithms, Dynamic Memory, Pointers

# Sprint 2

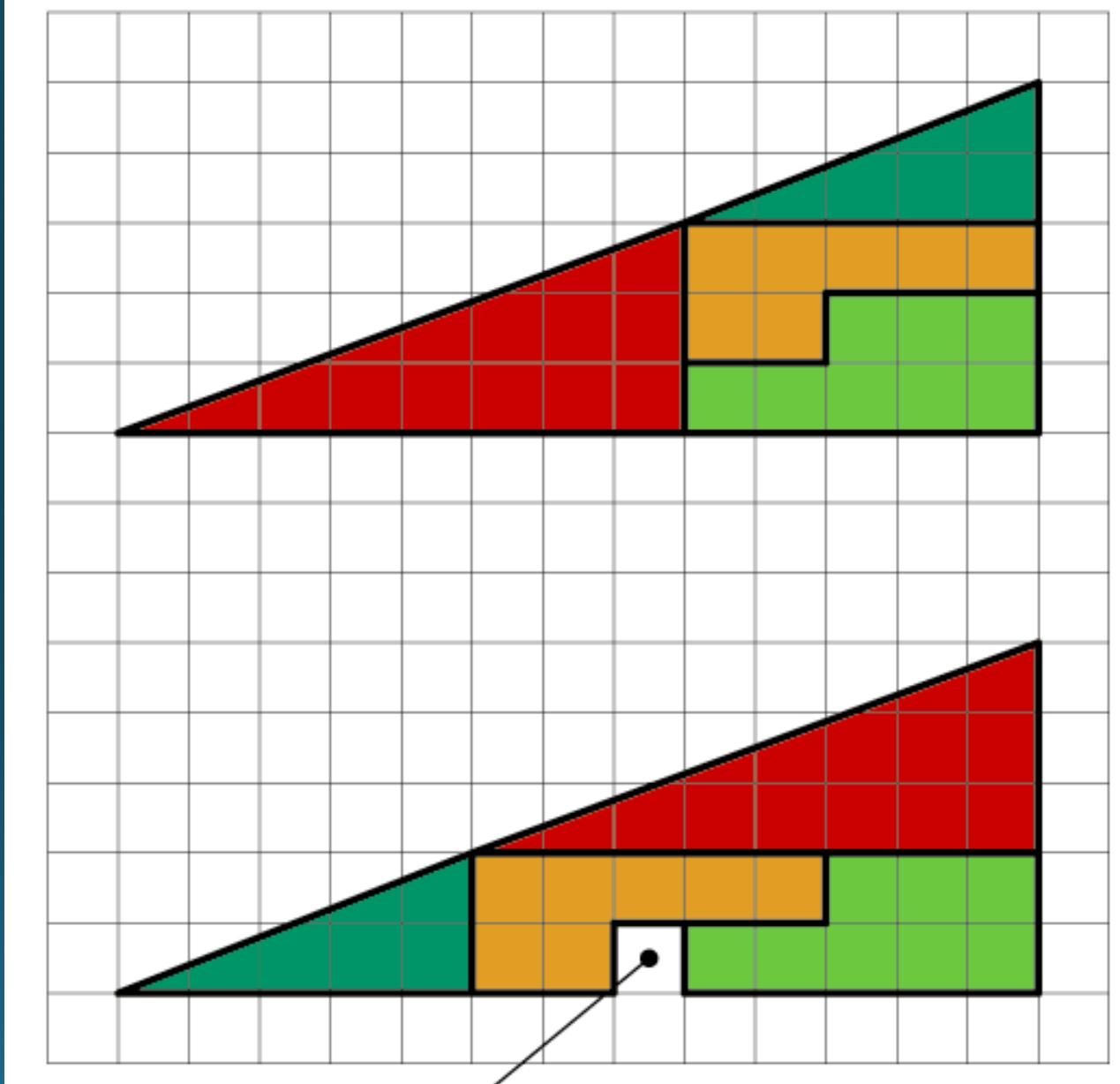
....

# Ready

....

# GO!

# Howwww ??



# Types

- ❖ Set up a template for a variable
- ❖ No memory allocation
- ❖ Not scoped

# Variables

- ❖ Set up a template for a variable
- ❖ Memory **is** allocated
- ❖ Scoped to the module

# Memory Allocation

# Static Data

- ❖ Fixed bits, create **safe environment!**
- ❖ Amount of bits used known **before run time**
- ❖ Scoped to the module
- ❖ Risk of wasting space or coming too short

# Dynamic Data

- ❖ Data can grow or shrink
- ❖ Amount of bits used known at running time
- ❖ Allocation and de-allocation when convenient
- ❖ Flexible but need to keep track of size and location

# Memory Structure

- ❖ Data can grow or shrink
- ❖ Amount of bits used known at running time
- ❖ Allocation and de-allocation when convenient
- ❖ Flexible but need to keep track of size and location

# Pointers

- ❖ Variables that store address
  - ❖ \*
  - ❖ &
- ❖ type \*var

# Pointers

```
int myInt = 5;  
cout<<myInt<<endl;
```

```
int *myPtr = myInt;  
cout<<myPtr<<endl;
```

.....

# Pointers

Address	Value	Variable
0xFFFF		
0xFFE		
0xFFD		
0xFFC		
0xFFB	0xFF	ptrX
0xFFA	0x01	
.		
.		
.		
0xFF04	0xFF	
0xFF03	0x0A	
0xFF02	0x01	X
0xFF01	0x05	
0xFF00		

# Lecture Quiz

*Key: ziujq2*