See MIPS Run

Yo

Objectives

- Introduction to programming with the MIPS instruction set
    - We will not be watching millions of instructions per second on the screen
- Explain the format of instructions
- Convert simple C code to MIPS code
    - Example 1: Variables and simple arithmetic
    - Example 2: Comparisons and branching
    - Example 3: Working with Main Memory, comparisons, and address offsets (with arrays)

Notes

- These examples may not be the most efficient
  - There are multiple ways to solve the same problem, some are more optimized than others
- This is strictly a programming exercise
  - We will not review MIPS registers, datapath, etc.
  - You will learn about in Computer Organization and Architecture
- Examples and documentation are on my Github
- https://github.com/danburkhol

Feel free to ask questions at any time

Instruction Format

	Operation	Destination	Source	Source 2 (or Address / Immediate)	Translation
R-Format (Arithmetic):	add	$s1	$s2	$s3	S1 = S2 + S3
I-Format (Immediate):	addi	$s1	$s2	100	S1 = S2 + 100
J-Format (Jump):	j			2500	Jump to Address 2500

```
1 // Example 1
2 // Declaring variables & Simple Arithmetic
3 int a = 6;
4 int b = 9;
5 int c = 0;
6
7 c = a + b;
```

```
2 # Example 1
3 # Declaring Variables & Simple Arithmetic
4         addi    $t0, $zero, 6       # init a = 6
5         addi    $t1, $zero, 9       # init b = 9
6         addi    $t2, $zero, 0       # init c = 0
7
8         add     $t2, $t0, $t1       # c = a + b
```

```
1  // Example 1
2  // Declaring variables & Simple Arithmetic
3  int a = 6;
4  int b = 9;
5  int c = 0;
6
7  c = a + b;
8
9  // Example 2
10 // Comparison
11 // Loops and branching
12 while (c < 100) {
13     c = c++;
14 }
15
```

```
# Example 1
# Declaring Variables & Simple Arithmetic
        addi    $t0, $zero, 6       # init a = 6
        addi    $t1, $zero, 9       # init b = 9
        addi    $t2, $zero, 0       # init c = 0

        add     $t2, $t0, $t1       # c = a + b

# Example 2
WHILE   slti    $t3, $t2, 100       # t3 = (t2 < 100) : 1 or 0
        beq     $t3, $zero, END     # branch to END if t4 == 0
        addi    $t3, $t3, 1         # Add immediate 1, t3++
        j WHILE                     # Jump back to WHILE Loop
END

```

```
16 # Example 3
17 # Comparison, arrays, and offset
18 # Say we're given array of integers named data[]
19 # Let's sort the first two items
20 if (data[0] > data[1]) {
21     int temp = data[1];
22     data[1] = data[0];
23     data[0] = temp;
24 }
25
```

```
15
16 # Example 3
17 # Given array base address is at register $s0
18 # Load the two data items from the array
19
20         lw      $t0, 0($s0)          # Fetch data[0] to register $t0
21         lw      $t1, 4($s1)          # Fetch data[1] to register $t1.
22                                      # 4 byte (32-bits) offset per item
23         slt     $t2, $t1, $t0        # t2 = (data[1] < data[0]) : 1 or 0
24         beq     $t2, $zero, END      # branch to END if t2 == 0
25         sw      $t1, 0($s0)          # data[0] = data[1]
26         sw      $t0, 4($s0)          # data[1] = data[0]
27 END
28
```