A dog in Melbourne with an embarrassing background story

COMP1511 Week 4!

H13A: 1pm – 4pm

Tutors: Me + Vivian Zheng

My GitHub:

https://github.com/william-o-s/unsw_comp1511_tutoring

Course Homepage:

https://cgi.cse.unsw.edu.au/~cs1511/23T3/

Please remember Census Date!

https://www.student.unsw.edu.au/dates

The Agenda

Arrays Practice (20 mins)

What are these? Have you heard these names before?

```
int array[3] = { 1, 2, 3 };
double list[3] = { 1.0, 2.0, 3.0 };
char collection[3] = { 'a', 'b', 'c' };
```

Looped Scans Part 1 (15 mins)

In groups, let's tackle a scanning problem...

- Problem:
  Scan temperatures into an array for 7 days. Find the highest temperature.
- What steps could you take?
  - Initialise an array
  - `while` loop using `scanf`
  - Insert into the array
  - Loop through array & track largest value
- Write some pseudocode / draw a flowchart!

Looped Scans Part 2 (10 mins)

Before diving into more looped `scanf`, let's cover input

Functions Practice (15 mins)

First, check out this `struct`!

```
 8   struct colour {
 9       int red;
10       int green;
11       int blue;
12   };
```

Kahoot!

What are these? Have you heard these names before?

```
int array[3] = { 1, 2, 3 };
double list[3] = { 1.0, 2.0, 3.0 };
char collection[3] = { 'a', 'b', 'c' };
```

In groups, fill in the blanks to print every array element

```
1    int array[3] = { 1, 2, 3 };
2
3    int i = 0;
4    while (i <  ) {
5        printf("%  ",  );
6        i++;
7    }
8
```

Did you get them all?

```
1    int array[3] = { 1, 2, 3 };
2
3    int i = 0;
4    while (i < 3) {
5        printf("%d", array[i]);
6        i++;
7    }
8
```

Let's try out the 'Odd Only' task first

1. Create an integer array with at least 5 elements
2. Create a **`while`** loop which loops through every element of the array.
3. Write an **`if`** statement which adds 1 to each **even** value. Do this within the **`while`** loop.
4. Write another **`while`** loop which goes through the array *with a different iterator* (i.e. if you used **`i`** last time, use **`j`**)
5. Print out the values in the array.

Now, let's try the other tasks

Copy Array

1. Create an array of **`double`**s with 3 elements, each with a non-zero value.
2. Create another array of **`double`**s with 10 elements where every element initialised to **`0.0`**.
3. Create a **`while`** loop that loops through every element of the first array.
4. Copy the elements of the first array into the second array (leave 0's at the end)
5. Create a **`while`** loop that prints out all the elements of the second array.
6. Go join other teams, and sit with their groups to help them finish.

Largest Character

1. Create a **`char`** array with exactly 8 elements.
2. Create a **`char`** variable called **`largest_character`**, equal to the first character of the array.
3. Create a **`while`** loop to loop through the **`char`**acter array.
4. Create an **`if`** statement to check if the current character has a higher ASCII value than **`largest_character`**
5. Print out the largest character you've found.
6. Ensure your code would pass the **`1511 style`** command
7. Go join other teams, and sit with their groups to help them finish.

In groups, let's tackle a scanning problem…

- Problem:
  Scan temperatures into an array for 7 days. Find the highest temperature.
- What steps could you take?
  - Initialise an array
  - **`while`** loop using **`scanf`**
  - Insert into the array
  - Loop through array & track largest value
- Write some pseudocode / draw a flowchart!

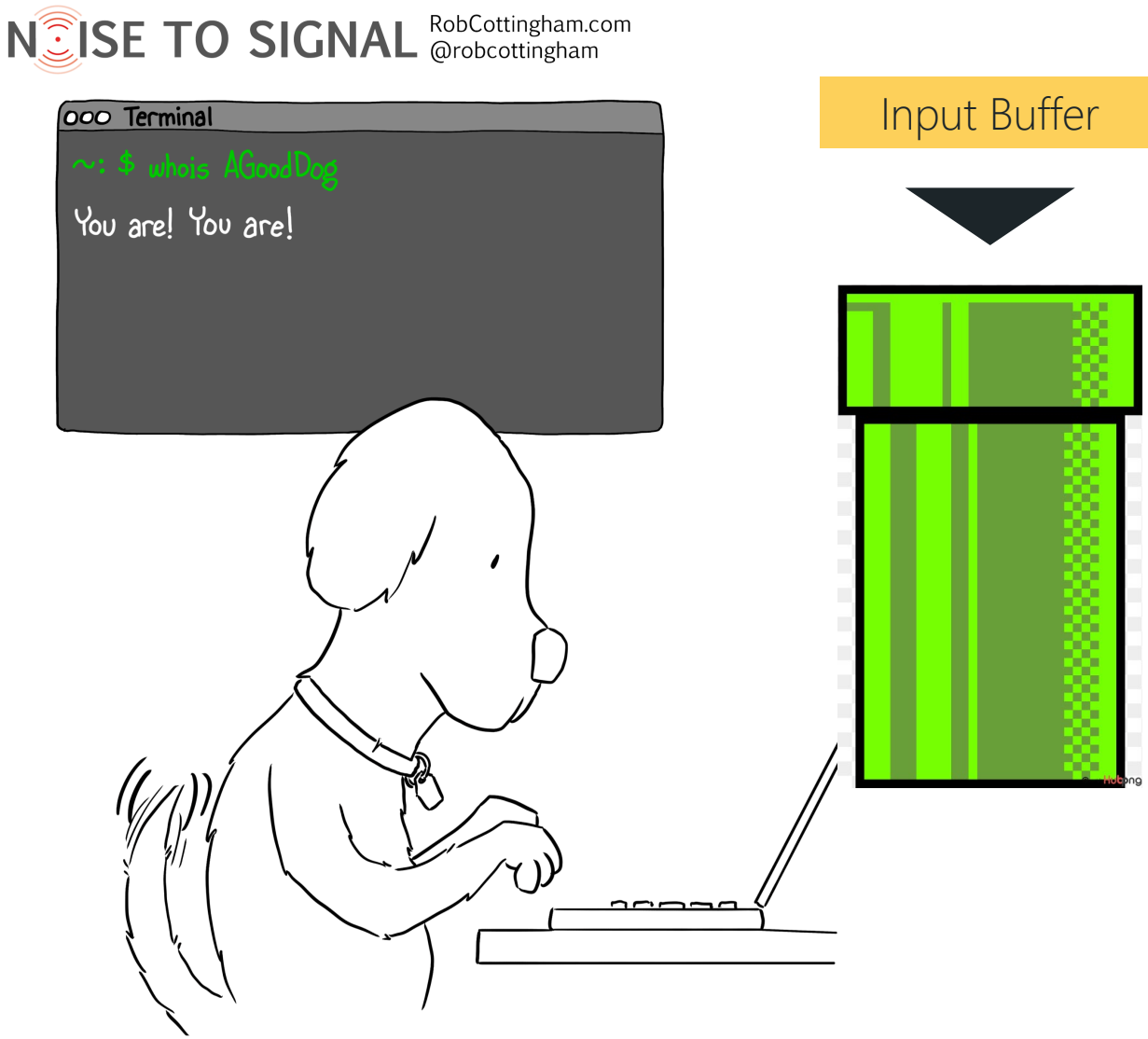
Before diving into more looped `scanf`, let's cover input

Before diving into more looped `scanf`, let's cover input

Input

"comp1511"

Before diving into more looped scanf, let's cover input



Input

'c' → 'o' → 'm' → 'p' → '1' → '5' → '1' → '1'

Before diving into more looped scanf, let's cover input

scanf

Input

'c' → 'o' → 'm' → 'p' → '1' → '5' → '1' → '1'

First, check out this `struct`!

```
 8   struct colour {
 9       int red;
10       int green;
11       int blue;
12   };
```

Second, check out this function!

```
 8   struct colour {
 9       int red;
10       int green;
11       int blue;
12   };
```

```
19   struct colour make_colour(int red, int green, int blue) {
20       struct colour new_colour;
21
22       new_colour.red = red;
23       new_colour.green = green;
24       new_colour.blue = blue;
25
26       return new_colour;
27   }
```

Third…let's Kahoot!