say nothing + survey + split:

Identify his last name

identify the labrador retriever

COMP1511 Week 2!

Operators, Data Types, Branching

The Agenda

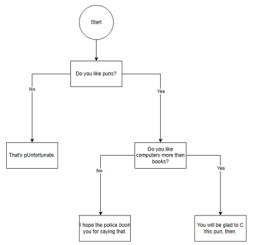
Calculations in C

Let's play: Operator Bingo!

Arithmetic	+ (add)	- (subtract)	* (multiply)	/ (divide)	% (modulo)	
Logic	&& (AND)	|| (OR)	! (NOT)			
Comparison	< (less than)	> (more than)	<= (less than or equal)	>= (more than or equal)	!= (not equal to)	== (is equal to)
Not currently covered	++ (increment)	-- (decrement)				
Not in C	^ (not exponentiation)	x & y (bitwise AND)	x | y (bitwise OR)	x ^ y (bitwise XOR)		

Diagramming before C

What's in this chart?



Let's code up a leap year calculator:

- Years divisible by 4 are leap years. (e.g. 1904 was a leap year)
- Except, years divisible by 100 are not leap years. (e.g 1900 was NOT a leap year)
- Except, years divisible by 400 are always leap years. (e.g. 2000 was a leap year)

note:
minimum of 3 group activities per tutorial

Data Types in C

What data types have we seen? What's their differences?

```
// Some data types
char character_type;
int integer_type;
double double_type;
```

```
// What is returned by the following?
int division_example = 10 / 5;
char character_example_one = 'a' + 1;
int character_example_two = 'a' + 1;
```

What is the mathematical, sensisible, and 'C' answer for these?

- `(7 / 2)`
- `(3.0 / 2) + 1`
- `'a' + 5`
- `'F' - 'A' + 'a'`

Programming in C

Let's code this!

In this activity, you'll be writing the following program.

It should:

- Scan in two integers (`a` and `b`).
- If the first integer is less than the second, print out a short error message **using a procedure**.
- If the second integer is 0, print out a different short error message.
- If the first integer is larger than the second, prints `a / b` and `(a * 1.0) / (b * 1.0)` .

```
// C style pseudocode example.
// Prints out "Hurrah!" if the entered number is 5

int n = 0
print "Enter a number"
scan a number into n
if (n == 5) {
    print "Hurrah!"
}
```

Let's play: Operator Bingo!

Arithmetic	+ (add)	- (subtract)	* (multiply)	/ (divide)	% (modulo)	
Logic	&& (AND)	|| (OR)	! (NOT)			
Comparison	< (less than)	> (more than)	<= (less than or equal)	>= (more than or equal)	!= (not equal to)	== (is equal to)
Not currently covered	++ (increment)	-- (decrement)				
Not in C	^ (not exponentiation)	x & y (bitwise AND)	x | y (bitwise OR)	x ^ y (bitwise XOR)		

What data types have we seen? What's their differences?

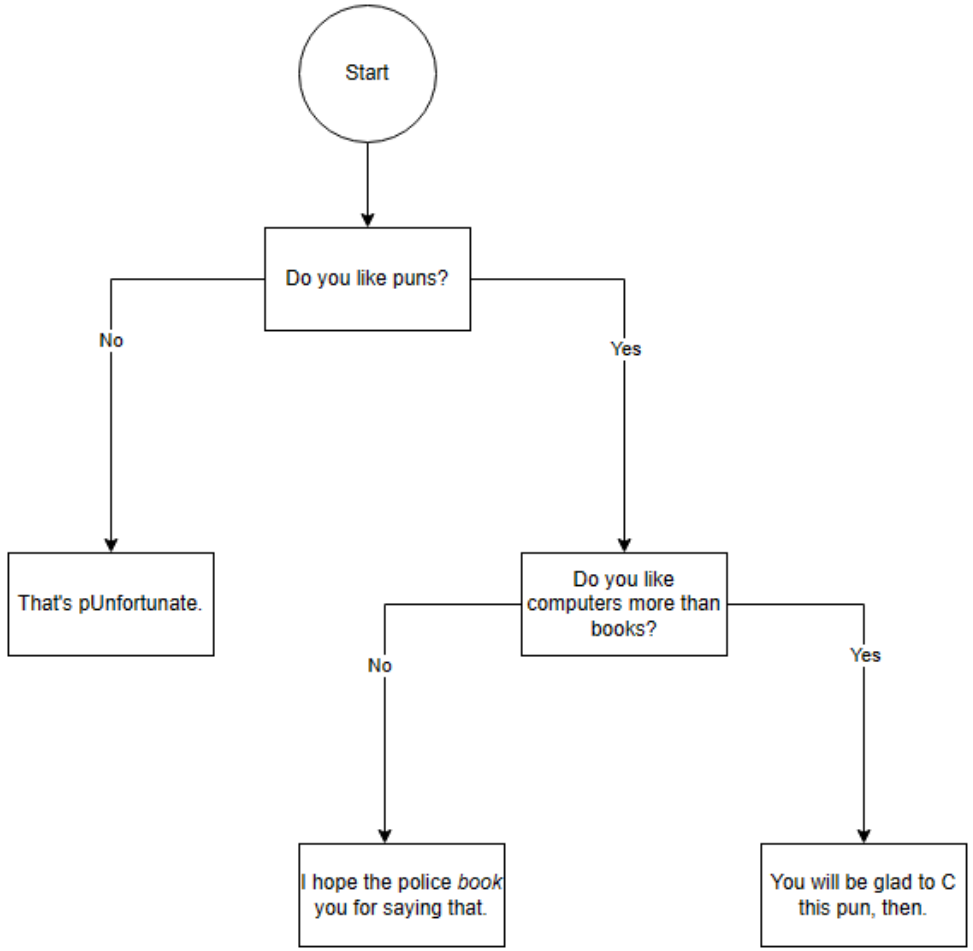
```c
// Some data types
char character_type;
int integer_type;
double double_type;
```

```c
// What is returned by the following?
int division_example = 10 / 5;
char character_example_one = 'a' + 1;
int character_example_two = 'a' + 1;
```

What is the *mathematical, sensisible, and 'C' answer for these?*

- `(7 / 2)`
- `(3.0 / 2) + 1`
- `'a' + 5`
- `'F' – 'A' + 'a'`

What's in this chart?



Let's code up a leap year calculator:

- Years divisible by 4 are leap years. (e.g. 1904 was a leap year)
- Except, years divisible by 100 are not leap years. (e.g 1900 was NOT a leap year)
- Except, years divisible by 400 are always leap years. (e.g. 2000 was a leap year)

Let’s code this!

In this activity, you'll be writing the following program.

It should:

- Scan in two integers (`a` and `b`).
- If the first integer is less than the second, print out a short error message **using a procedure**.
- If the second integer is 0, print out a different short error message.
- If the first integer is larger than the second, prints `a / b` and `(a * 1.0) / (b * 1.0)` .

```
// C style pseudocode example.
// Prints out "Hurrah!" if the entered number is 5

int n = 0
print "Enter a number"
scan a number into n
if (n == 5) {
    print "Hurrah!"
}
```