Module 4 Part 1: Data Types and Expressions (int)

Each Programming Language has 3 Fundamental Constructs:
1. Data
2. Expressions
3. Control Flow
   - In Order to learn and master a language you must know these 3 Constructs.

The Default "flow" of a program is called: Sequential Flow
- Sequential Flow: Control Flow where a program executes the code one line after another.

- Data: In C++ Data is "strongly" typed, where a piece of Data must have a specified "Type".
  - Example: int num1; //num1 data with type "int" for interger

- Expressions: Come in all forms such as Arithmatic Expressions (+, -, /, x, %) or Assignment (sum = num1 + num2), I/O ("cout <<", "cin >>")

Data Types

- Several Data Types used in C++ :
  - int
  - float
  - double
  > -> (Use to represent "real numbers" w/ fractional parts)
  - char -> (Used for characters)
  - string -> (String for textual Information)
  - boot -> (Boolean Values "True" and "false")

The "int" Data Type

- C++ Literals:
  - Data can come in 2 forms: Variables or Constants
  - Variables can change their values.
  - Constants are values that are constant and cannot be changed
  - Every Data has a type of its own

- So we can have variables that have a type
  (i.e. "int x" or "double y")

- Or constants such as C++ Literals
  (i.e. "abc", "6", "5.3")
  - These values don't need to be defined because the compiler already recognizes them (Built inside the language)

- But we can also define our own constants (Programer Defined)
  (i.e. "const int MAX = 5;", constant of type int
  constant of type int named MAX set to value 5)
  - Once MAX is set to 5 IT CANNOT CHANGE

- The compiler recognizes numbers written in their NON-FRACTIONAL decimal representation (i.e. 3, 7, -6, 23547)
- The compiler would recognize these numbers as intergers

The "int" Data Type

- Mathematic Operators:
  - Using Mathematic Operators we can create expressions
  (i.e plus "+"): We can add 2 integers and create a larger expression
  * - Watch Video for example Reference
  - "+" can come inbetween two "int's" to create a compound expression
  - We can then print this expression OR (cout << x+2;)
  - We can set a variable to this expression (y = x+2;)
  ↳ This is done with assignment "="

- When an assignment is evaluated, the "right" is evaluate and assigned to the "left"

- There are also more operators
  - multiplication, subtraction, divider

- Divider ("/") is tricky

The "int" Data Type

- Div ("/") and Mod ("%")
  - If we do 13 div 5 (13/5) we get 2
  - Div ("/") means how many FULL TIMES 5 goes into 13 which is 2
    13 div 5 = 2

- However 13 mod 5 (13%5) gives us 3
  - Mod ("%") means whats the REMAINDER when 13 is divided by 5
    13 mod 5 = 3

- In C++, we dont use "div" and "mod" in their textual writing
  - There are specific symbols:
    - Div ("/")
    - Mod ("%")

- If we do "x/2" and x is an "int" of value 5 we get 2
- If we do "x%2" and x is an "int" of value 5 we get 1

Assignment Operator (=)

- Assignment Operator assigns a variable with a value, by first evaluating the right side (2) and assigning it to the left (x) to give (x = 2)

- We can also assign variables that have value to other variables i.e. (y = x) this would give y the value of x which is 2.

- We can further expand these expressions such as
  - x = 3 "x" has value "3"
  - y = x "y" has value "x"
  - y = x = 4 "y" has value "x" which has value "4"

The "int" Data Type Summary

- Used to represent "Whole" Numbers
- Each have a fixed size of 4 bytes (1 byte = 8 bits)
- Data is represented using a "32-bit 2's complement") method
- C++ has built in data that's considered integers = C++ Literals
- We can create arithmetic operators using integers in between them