Programming – 23.09.19

* Directory of lecturer emails in the student handbook
* “Programmers don’t like writing documentation, poor at communication, this needs to be improved”
* “//” is a comment, this is a statement which is not interpreted by the code. Used to annotate code. – single line comment
* “/\* insert comment

Insert comment

Insert comment

\*/” – Multi line comment “/\* being the start, \*/ for the end.”

* When you start off a program, start with a comment stating the programs description. This states the programs intended purpose, or rather a summary of it’s intended purpose. Under this write “Author” and your name, under that, place a “date” of the program’s creation or last modification.
* **^ There is a mark for this in lab tests/assignments**
* Data Types:
* Data types known so far (check other notes for descriptions). Int, float, char.
* There are three different types of Integer (Int):
* Short, which has a range of -32000 to +32000, this can be shown by station “short X” – X being the number.
* Int, which has a range of +/- 2 million.
* Long, which goes from a +/- BIG RANGE
* -
* **A short used the least amount of memory, a int is the middle, a long is the most.**
* Efficiency is key, use the smallest int possible.
* Some applications have *very small* amounts of memory available, this means memory optimisation is key.
* %ld = log
* %d = int/short/long
* Float = data type for storing numbers as decimals
* Types of floats:
* Float, occupies 4bytes, +/- 2million
* Double, occupies 8bytes, +/- BIG RANGE
* API = application program interface = set of already written functions which can be accessed.
* -
* Char = (refer to previous notes, words basically)
* Boolean = binary variable, can only hold a 1 or 0.
* %B is the delimiter for a Boolean.
* -
* Simple arithmetic operations:
* *Example code – initialising and adding a value in one line*

int num1 = 5;

int num2 =10;

* *Initialising multiple int’s at once. For the second one, adding an “int” will not work.*

Int num1, num2 = 10;

num1 = num2 = 10;

* Addition “+”
* Subtraction “-“
* Multiplication “\*”
* Division “/”
* Modulus “%” – Used for giving the remainder of an arithmetic operation.
* *Example code – Arithmetic and variables, adding a variable “num2” to a number*

int num1, num2;

num1 = 0;

num2 = 10;

num1 = num2 +4;

printf (“num1 contains %d \n”, num1)

* Arithmetic MUST be on the right-hand side, so X = Y+Z. It will NOT work if you do Y+Z = X.
* *Example code – different operation, continuing from the last arithmetic. Showing that num2 doesn’t change, but num1 does.*

num1 = num2 – 10;

printf (“num1 contains %d \n”, num1);

* *Example code – showing example of a multiplication operation, continuing from last code. This is the reason for num2 being redefined.*

num2 = 2;

num1 = num2 \* 5;

* *Example code – division operation*

num1 = num2 / 2;

printf (“num1 contains %d \n”, num1);

* *Example code – modulus operation*

num2 = 3;

num1 = num2 % 2;