Programming – 25.09.19

* “unsigned int num;” = this takes away the negative numbers in a variable, meaning the variable can only be assigned a positive number.
* NOTE: There is no way to do the opposite. I.E ONLY have negative numbers.
* NOTE: if a negative number is placed in an unsigned int after it is called, an exception will be thrown. I.E it will fail to compile, or if the program is running. The program will crash.
* FLOAT and FLOATING POINT the same
* REMEMBER! After a division or arithmetic operation, if the new number is a decimal, and it is placed in an “int” the decimal nature of the value will be lost. I.E “4.9” is placed in an INT the “4.9 part of the “4.9” is lost, leaving “4”
* EXAMPLE CODE: - To explain the above REMEMBER ^

#include (stdio.h)

int main()

{

Int var1 = 10;

Float var2 = 2.5;

float var3;

int var4;

float var5;

var3 = var1 / var2;

var3 = var1 / var2;

var5 = var1 / var4;

printf(“var3 is %f, var4 is %d, var5 is %f,” var3, var4, var5)

return 0;

}

* If you divide an integer by an integer, the answer will always be an integer.
* Solution to bug 1: Divide an integer by an integer
* Solution 2: “Casting (Cast)” – “var5 = (float)var1 / 4; “
* **ONCE INITIALISED, A DATA TYPE CANNOT BE CHANGED**