Programming – 01.10.19

* EXAMPLE CODE –

// Program to enter any character

#include (stdio.h)

Int main()

{

Char ch;

Printf(“Enter in any character \n”);

Scanf(“%c”, &ch);

Printf(“You entered %c”, ch);

Return 0;

}

* %s = delimiter for string
* %1s = 1 letter string, meaning it will only display (in print) or will allow input (in scanf) of one letter.
* ^ This solution, saving as a string, will sanitise white space characters.
* “getchar()” is a depreciated method, not used anymore and some compilers may throw an warning. “Don’t worry about this warning”
* EXAMPLE CODE:
* “Getchar” is MUCH faster then a “Scanf” – More efficient
* There is no getint, or getvar – use scanf
* Printf(“\n You entered %c”, ch); = putchar(Ch);
* PUTCHAR is better for single words, or printing answers. Printf is better for longer sentences.
* GETCHAR and PUTCHAR only work for ONLY character variables
* EXAMPLE CODE

#include (stdio.h)

Int main()

{

Char ch;

Printf(“enter any character\n”);

Ch = getchar();

* You CANNOT put anything in anything except delimiters into the scanf
* CONTROL STATEMENTS
* If statement sudocode

if (condition)

{

Statements, etc

}

* If the condition is true, all statements inside will be run. If false, they will be skipped.
* EXAMPLE CODE

Int balance;

Balance = 100;

If (balance < 100)

{

Printf(“Your balance is below 100”);

}

* “ ; “ Ends that line, so in an if statement. A ; after the if statement will invalidate the “if” statement, meaning all the code after it will always execute.
* Compilers will ONLY catch syntax errors, not logical errors.
* Note: If you forget the {} the statement will only process the first line after the if statement for its process. Anything after that line will not be bound by the if statements rules.
* Even if not strictly required, always include the {}
* test