\’ single quote

\” double quote

\\ backslash

\0 null value

\b backspace

\n new line

int vari1= 5;

float vari2 = 3.2;

string vari3 = “hello”;

char vari4 = ‘a’;

bool vari5 = True;

const double GST\_FACTOR = 1.15;

int vari9 = Convert.ToInt32(temp);

C.WL($”Her name is {name}.”);

{4238.8:c2} => $4,238.80

{42:d5} => 00042

{4238.8:e2} => 4.24E+003

{4238.8:f4} => 4238.8000

{4238.8:n3} => 4,238.800

{0.123:p1} => 12.3%

using System.Threading; (@ top)

Thead.sleep(1000) = sleep 1 sec

Random rand = new Random();

randnum = rand.Next(a,b)

a inclusive b not inclusive

string[] myBooks = new string[#ofitems]

always start at ith place 0

moBooks[3] = “a book name”

C.WL(myBooks[12])

string name = “Hamza”

C.WL(name[3]) = ‘z’ (read only)

name=name.Replace(‘m’, ’n’) = “Hanza”

Replaces all ‘m’ in string

name.Contains(“Ha”) = True

name.IndexOf(‘z’) = 3

name.ToLower() = “hamza”

name.ToUpper() = “HAMZA”

string sample = “brown fox”

sample.Replace(“brown”, “red”) = “red fox”

sample.Insert(0, “quick”) = “quick brown fox”

sample.Remove(3, 2) = “bro fox” (start, #remove)

sample.Substring(3,2) = “wn”

sample.Substring(3) = “wn fox”

sample.IndexOf(“orange”) = -1 (not exist)

sample.LastIndexOf(“brown”) = 4

string[] vari = sample.Split(‘w‘) = [“bro”, “n fox”]

sample.Trim() = “brownfox”

“words”.PadRight(10) = “words “ (10 char total)

“word”.PadLeft(8) = “ word” (8 char total)

**MATH**

Division eg. 11/4 = 2 cause it says how many

times can 4 fit and throws away the rest

Modulus eg. 11%4 = 3 ignore how many times

it goes in, give me the remainder

> greater than

< less than

>= greater than or equal to

<= less than or equal to

== equal to

!= not equal to

vari++ = increase by 1

vari-- = decrease by 1

Math.PI = pi

Math.E = e

Math.Abs(a) = absolute

Math.Ceiling(a) = round up to int

Math.Floor(a) = round down to int

Math.Max(a,b) = highest (only 2 inputs)

Math.Min(a,b) = lowest

Math.Pow(a,b) = a to power of b

Math.Round(a,b) = rounds a to b decimal #

Math.Sqrt(a) = square root of a

float avg = (int1 + int2 + int3) / (double) count;

makes output the type in brackets

**Conditionals and decisions**

(a<b) && (a<c) a<b AND a<c

(a>b) || (a>c) a>b OR a>c

if (a<b) {…..;}

else if (b<a) {….;}

else {….;}

switch (int or char or string) {

case [value]: do thing; break;

case [value]: do thing; break;

default: do thing; break;

while (num < 10) {must change num in here;}

do {….;

} while (num < 5);

int i; (declare before loop to keep the vari)

for (i=1; i<10; i++) {…..;}

for (char letter=‘a’; letter<’h’; letter++)

int[] scores = {1,2,3,4,5}

foreach (int score in scores)

{total = total + score}

foreach (char I in string)