<C# Programs>

First Concept

string x = Console.ReadLine();

string message = (x == "1") ? "true" : "false";

Console.WriteLine("You entered {0} therefore your result is {1}", x, message);

Console.ReadLine();

Second Concept of Array

string[] x = new string[] {"Saad", "Asif", "Ahmad", "Anas" };

Console.WriteLine("Length of Array is: " + x.Length);

foreach (string item in x)

{

Console.WriteLine(item);

}

Reverse Array Program

string zig = "Hello World!";

char[] z = zig.ToCharArray();

Console.WriteLine("Length of zig is: " + z.Length);

Array.Reverse(z);

foreach (char reverse in z)

{

Console.WriteLine(reverse);

}

Concept Of Methods

static void Main(string[] args)

{

string x = Console.ReadLine();

Console.WriteLine(string.Format(revers(x)));

Console.ReadLine();

}

private static string revers(string x)

{

string zig = x;

char[] z = zig.ToCharArray();

Console.WriteLine("Length of zig is: {0} " , z.Length);

Array.Reverse(z);

return string.Concat(z);

}

# NOTE:

# You can also use your language's string concatenation operator, such as + in C#, or & and + in Visual Basic), to concatenate strings.

# To create an overloaded method:

# You create two or more methods with the same name, but with different number of input parameter and / or different data types.

# For taking integer input:

//This syntax will use

int x = int.Parse(Console.ReadLine());

Bool and While Iteration Concept

static void Main(string[] args)

{

bool x = true;

while (x)

{

x = hi();

}

}

private static bool hi()

{

Console.Clear();

Console.WriteLine("1) Number Game!");

Console.WriteLine("2) Guessing Game!");

Console.WriteLine("3) Exit!");

int input = int.Parse(Console.ReadLine());

if (input==1)

{

number();

return true;

}

else if (input==2)

{

game();

return true;

}

else if (input==3)

{

return false;

}

else

{

return true;

}

}

private static void number()

{

Console.Clear();

Console.Write("Enter a number = ");

int y = int.Parse(Console.ReadLine());

int z = 1;

while (z<y)

{

Console.Write(z + "-");

z++;

}

Console.WriteLine("\n");

Console.ReadLine();

}

private static void game()

{

Console.Clear();

Console.WriteLine("Guessing Game!");

int guess = 0;

bool counter;

Random u = new Random();

int v = u.Next(1, 10);

do

{

Console.WriteLine("Enter any number between (1 to 10) = ");

int yu = int.Parse(Console.ReadLine());

guess++;

if (yu==v)

{

Console.WriteLine("Your answer is correct!");

Console.WriteLine("You have done it in {0} attempts.",guess);

counter = false;

}

else

{

Console.WriteLine("WRONG!");

counter = true;

}

} while (counter==true);

Console.ReadLine();

}

Working with Strings

//string myString = "My \"so-called\" life";

//string myString = "What if I need a\nnew line?";

//string myString = "Go to your c:\\ drive";

//string myString = @"Go to your c:\ drive";

//string myString = String.Format("{1} = {0}", "First", "Second");

//string myString = string.Format("For Currency {0:C}", 123.45);

//string myString = string.Format("For Arrange numbers: {0:N}", 1234567890);

//string myString = string.Format("Percentage: {0:P}", .123);

//string myString = string.Format("Phone Number: {0:(###) ###-####}", 123456789012);

//string myString = " That summer we took threes across the board ";

//myString = myString.Substring(6, 14);

//myString = myString.ToUpper();

//myString = myString.Replace(" ", "--");

//myString = myString.Remove(6, 14);

/\*

myString = String.Format("Length before: {0} -- Length after: {1}",

myString.Length,

myString.Trim().Length);

\*/

/\*

string myString = "";

for (int i = 0; i < 100; i++)

{

myString += "--" + i.ToString();

}

\*/

StringBuilder myString = new StringBuilder();

for (int i = 0; i < 100; i++)

{

myString.Append("--");

myString.Append(i);

}

Console.WriteLine(myString);

Console.ReadLine();

Working Dates and Times

DateTime myValue = DateTime.Now;

//Console.WriteLine(myValue.ToString());

//Console.WriteLine(myValue.ToShortDateString());

//Console.WriteLine(myValue.ToShortTimeString());

//Console.WriteLine(myValue.ToLongDateString());

//Console.WriteLine(myValue.ToLongTimeString());

//Console.WriteLine(myValue.AddDays(3).ToLongDateString());

//Console.WriteLine(myValue.AddHours(3).ToLongTimeString());

//Console.WriteLine(myValue.AddDays(-3).ToLongDateString());

//Console.WriteLine(myValue.Month);

//DateTime myBirthday = new DateTime(1969, 12, 7);

//Console.WriteLine(myBirthday.ToShortDateString());

DateTime myBirthday = DateTime.Parse("12/7/1969");

TimeSpan myAge = DateTime.Now.Subtract(myBirthday);

Console.WriteLine(myAge.TotalDays);

Console.ReadLine();

## Which data type represents the elapsed time between two specific DateTime objects?

## TimeSpan.

Understanding Classes

class Program

{

static void Main(string[] args)

{

car x = new car();

x.model = "BMW";

x.year = 2018;

Console.WriteLine(x.model);

Console.WriteLine("{0:C}" , x.value());

Console.ReadLine();

}

}

class car

{

public string make { get; set; }

public string model { get; set; }

public string color { get; set; }

public int year { get; set; }

public decimal value()

{

decimal carvalue;

if (year>1990)

{

carvalue = 10000;

}

else

{

carvalue = 2000;

}

return carvalue;

}

}