**Seminario de Lenguajes (.NET) Práctica 1**

1. Consulte en la documentación de Microsoft .Net Framework sobre System.Console. WriteLine(),System.Console.Write() y System.Console.Readkey() Escriba un programa que imprima en la consola la frase “Hola Mundo”.
2. Consulte en la Documentación de Microsoft .Net Framework SDK por las secuencias de escape \n, \t \" y \\ Escriba un programa que las utilice para imprimir distintos mensajes en la consola.

-El \n es como el enter, lo que se escriba a continuación se escribe en la línea de abajo

-El \t es como usar un tab

-El \" sirve para imprimir una comilla

1. ¿Cuál es el resultado de ejecutar la siguiente instrucción?

Console.**Write**("\a");

Se reproduce una vez un sonido

y la siguiente?

Console.**Write**("\a\a\a\a\a\a");

Se reproducen tantos sonidos como \a

1. Compile y ejecute el siguiente programa

class Program

{

**public** static void **Main**(string[] args)

{

**for** (**int** i=1;i<=10 ;i++ )

{

System.Console.**WriteLine**("Hola Mundo!\a");

System.Threading.Thread.**Sleep**(500);

}

System.Console.**WriteLine**("Presione una tecla para continuar");

System.Console.**ReadKey**(**true**);

}

}

¿Qué ocurre si suprime la instrucción System.Threading.Thread.**Sleep**(500);? Observe en qué momento se visualiza la frase en la consola la frase "Presione una tecla para continuar".

Si se suprime esa instrucción se imprimen todos los hola mundo de corrido sin la espera de 500 ms.

1. Qué ocurre con la siguiente instrucción en tiempo de compilación:

Console.WriteLine("c:\documento.txt”);

¿Cómo puede solucionarlo?

Lo que ocurre es que el programa no compila debido a una secuencia de escape no reconocida. Para solucionarlo se reemplaza la línea por:

Console.WriteLine(@"c:\documento.txt");// usando el @ las sentencias de escape no tienen efecto

1. Escriba un programa que solicite al usuario ingresar su nombre e imprima en la consola un saludo personalizado utilizando ese nombre o la frase “Hola mundo” si el usuario ingresó una línea en blanco. Para ingresar un string desde el teclado utilice System.Console.ReadLine()
2. Idem. al ejercicio anterior salvo que se imprimirá un mensaje de saludo diferente según sea el nombre ingresado por el usuario. Así para “Juan” debe imprimir “¡Hola amigo! Me alegro de verte”, para “María” debe imprimir “Buen día señora”, para “Alberto” debe imprimir “Hola Alberto, que tenga usted un buen día”. En otro caso, debe imprimir “Buen día ” seguido del nombre ingresado o “¡Buen día mundo!” si se ha ingresado una línea vacía.

a) utilizando if ... else if

b) utilizando switch

1. Utilice System.Console.ReadLine() en un programa que lea por consola cadenas de caracteres, por cada string leído debe imprimir la cantidad de caracteres del mismo. El programa termina al ingresar la cadena vacía. (si **st** es una variable de tipo string, **st.Length** devuelve la cantidad de caracteres del string).
2. Qué hace la instrucción System.Console.**WriteLine**("100".Length); ?

Imprime 3, es decir la cantidad de caracteres que tiene el string que se imprime por pantalla. Similar al ejercicio anterior que el string estaba guardado en una variable

1. Sea st una viariable de tipo string correctamente declarada. ¿Es válida la siguiente instrucción: Console.**WriteLine**(st=Console.**ReadLine**());?

Si es válida. Imprime en pantalla exactamente lo ingresado por teclado. Es similar a un “echo”

1. Escriba un programa que lea dos secuencias de caracteres separadas por blanco que terminan con <ENTER>, y determine si son simétricas (Ej: 'abbccd' y 'dccbba' son simétricas).
2. Escriba un programa que imprima en la consola todos los números que sean múltiplos de 17 o de 29 comprendidos entre 1 y 1000
3. Compile y ejecute el siguiente programa.

**using** System;

class Program

{

**public** static void **Main**(string[] args)

{

Console.**WriteLine**("10/3 = " + 10/3);

Console.**WriteLine**("10.0/3 = " + 10.0/3);

Console.**WriteLine**("10/3.0 = " + 10/3.0);

**int** a=10,b=3;

Console.**WriteLine**("Si a y b son variables enteras, si a=10 y b=3");

Console.**WriteLine**("entonces a/b = " + a/b);

**double** c=3;

Console.**WriteLine**("Si c es una variable double, c=3");

Console.**WriteLine**("entonces a/c = " + a/c);

Console.**ReadKey**();

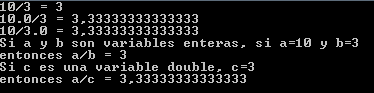
}

}

a) ¿Qué puede concluir respecto del operador de división “/”?

b) ¿Cómo funciona el operador + entre un string y un dato numérico?

La salida del programa es



1. Escriba un programa que imprima todos los divisores de un número entero ingresado desde la consola. Para obtener el entero desde un string st utilice Convert.ToInt32(st). (también puede utilizar int.Parse(st)).
2. Escriba un programa que calcule la suma de dos números reales introducidos por teclado y muestre el resultado en la consola (utilice Convert.ToDouble(st) o double.Parse(st) para obtener el valor real a partir del string st.
3. Escriba un programa que multiplique por 365 el número entero ingresado por el usuario desde la consola. El resultado debe ser impreso en la consola dígito por dígito, uno en cada línea, comenzando por el dígito menos significativo al más significativo.
4. Escriba un programa que asigne a una variable entera el resultado de la división *a*/*b*, siendo *a* y *b* dos valores reales ingresador por el usuario desde la consola.
5. Escriba un programa que pida un año por pantalla y diga si es bisiesto. Un año es bisiesto bajo si es divisible por 4 pero no por 100. Si es divisible por 100, para ser bisiesto debe ser divisible por 400.
6. Si a y b son variables enteras, identifique el problema (y la forma de resolverlo) de la siguiente expresión:

**if** ((b != 0) & (a/b > 5)) Console.**WriteLine**(a/b);

El problema es que si se ingresa b=0, se produce una excepción debido a que se realizan los dos términos del if, y el Segundo es erróneo al dividir por cero. La solución es en lugar de utilizar &, utilizar **&&**, de ese modo si el primero termino del if da falso, no se realiza la segunda evaluación, por ende no se produce la excepción.

1. Para cada una de las siguientes líneas de código verifique cuáles son permitidas por el compilador y en tal caso en qué estado quedan las variables involucradas en la declaración al ejecutarse:
2. **int** a, b, c;

Permitido

1. **int** a; **int** b; **int** c,d;

Permitido

1. **int** a=1; **int** b=2; **int** c=3;

Permitido, en cada variable queda el valor ingresado

1. **int** b; **int** c; **int** a=b=c=1;

Permitido, last res variables quedan con valor 1

1. **int** c; **int** a, b = c = 1;

NO PERMITIDO (uso de la variable no asignada a)

1. **int** c; **int** a=2, b = c = 1;

Permitido, con los siguientes valores a=2, b=1, c=1

1. **int** a=2, b, c, d = 2\*a;

NO PERMITIDO, (uso de las variables no asignadas b y c)

1. **int** a=2, **int** b=3, **int** c=4;

NO PERMITIDO, en b y 3 esta de mas el int

1. **int** a=2; b=3; c=4;

NO PERMITIDO, falta la declaracion de int en b y c

1. **int** a; **int** c=a;

NO PERMITIDO, le asigno a c la variable a que no tiene un valor

1. **char** c='A', string st="Hola";

NO PERMITIDO, conflito por como se declare el string

1. **char** c='A'; string st="Hola";

Permitido con lo asignado en cada variable

1. **char** c='A', st="Hola";

NO PERMITIDO, se quiere asignar un string a una variable char

1. ¿Cuál es el problema del código siguiente y cómo se soluciona?

**public** static void **Main**(string[] args)

{

**int** i=0;

**for**(**int** i=1;i<=10;){

Console.**WriteLine**(i++);

}

Console.**ReadKey**();

}

El problema es que dentro del for la variable de control declarada tiene el mismo nombre que la anterior (i), por lo tanto se declararia dos veces.

La solución es cambiar el nombre de alguna de ellas. El for parece que esta mal, pero esta bien y funciona ya que dentro del mismo se aumenta la variable de control.

1. Analice el siguiente código

**public** static void **Main**(string[] args)

{

**int** i=1;

**if** (--i==0){

Console.**WriteLine**("cero");

}

**if** (i++==0){

Console.**WriteLine**("cero");

}

Console.**WriteLine**(i);

Console.**ReadKey**();

}

Imprime

Cero

Cero

1

Esto se debe a que en el primer if primero hace el decremento y luego la comparación, por lo que da cero. En el segundo if primero hace la comparación que da cero y luego recién aumenta la variable, quedando finalmente con el valor 1

1. Investigue la estructura de directorios que crea el SharpDevelop para alojar las soluciones y proyectos que crea. ¿Dónde se genera el archivo ejecutable?

Crea una carpeta para cada solución, y dentro de la carpeta bin, crea una carpeta debug donde esta el ejecutable

1. Experimente con el menú “*Generar*” opción “*Ajustar configuración:*” del SharpDevelop eligiendo *Debug* y *Release*. Observe cómo esta elección afecta su respuesta anterior.
2. ¿Cuál cree que sea la diferencia entre los ejecutables generados bajo las opciones *Debug* y *Release*?