



Tarea (A) Programación Aplicada 1

- **Nombre:** Orlando Lora Ortega.
- **Matricula:** 2018-0068.
- **Profesor:** Enel Ramon Almonte Pichardo

Autoevaluación : Capítulo 1

¿Cuál es la historia del desarrollo de las aplicaciones para windows

Usando un compilador de C o C++ y de un Application Programming Interface (API) de windows era la única forma donde se podía programar en windows, el problema surgió al ver que el (API) de win32 es realmente amplia, y para facilitar esta programación surgen diversas opciones como (OWL) y (MFC), Creado por Microsoft.

MFC simplifica muchas cosas, pero algunos programadores no se sientían a gusto con el mismo. Otra opción fue Visual Basic, que trabaja por encima de Win32, pero su sintaxis se basa en el lenguaje Basic. Antes de la versión .Net, tenía limitaciones, ya que las (PO) no se podía llevar a cabo. Agregando a la lista de opciones, surge Java con su propio Framework.

Pero cada lenguaje tenía su compilador, era difícil compartir código. Para esto surge COM, un modelo que permite programar un componente en Visual Basic y otro puede tomarlo y utilizarlo. Esta programación también tenía complejidades, donde paso a ATL para ayudarlo. Con la filosofía de ordenar, facilitar y organizar el desarrollo en windows, surge .NET.

2-¿Qué problemas ayuda a resolver .NET?

Ayuda a no estar con preocupaciones por el uso de Memoria.
Ayuda a mantener un estándar por medio del (CTS), donde cada programador debe cumplir con la guía correspondiente para poder trabajar.

Por medio del (CTS) define como van a trabajar los tipos de su entorno.

El código puede ser optimizado por el compilador para la plataforma hacia donde se utilizará el programa, sin que se tenga que realizar cambios en el mismo.

3-¿Qué es un assembly?

Es lo que contiene el programa compilado como (CIL), y también información sobre los tipos que se utilizarán.

4-¿Cuál es la definición de CIL y CLR?

El CIL es un lenguaje intermedio que es leído y ejecutado por el runtime.

El CLR (Common Language Runtime) es un programa de ejecución común a todos los lenguajes.

5-¿ Describe que hace el CLR con el Assembly?

cuando se lleva a cabo la compilación, se genera un assembly. cuando se ejecuta, el CLR se encarga de ejecutarlo.

6-¿ Cual es el significado de JITler?

Es el compilador que manda a ejecutar lo que se ha compilado, en vez de volver a compilar.

7-¿ Mencione algunos compiladores de C# que podemos utilizar?

Compilador JIT o JITler, Compilador de C# express

8-¿ Qué es el CTS?

El CTS (Sistema de Tipos Comunes) es el encargado de verificar los tipos en su entorno.

9-¿ Porque .Net es multiplataforma?

Porque se puede ejecutar en muchas plataformas y si otra tiene el CLR o Runtime, nuestro programa se ejecutara.

10-¿ Qué otro lenguaje que use .Net existe?

Basic.

11-¿Cuál es la última versión de .Net?

4.6

12-¿Qué otro lenguaje que use .Net existe?

Basic.

Capítulo 2

1-¿Qué es un algoritmo?

Es un conjunto de pasos ordenados necesarios para llevar a cabo una acción.

2-¿Qué características tienen los algoritmos?

* Debe de tener un inicio y un Fin.

* Debe ser Preciso.

* Debe ser Determinado.

3-¿ Qué es una sentencia?

Las sentencias es el espacio donde se nos permite colocar instrucciones que nuestro programa pueda ejecutar.

4-¿ Cómo se finalizan las sentencias?

Con Puntos y Comas (;)

5-¿Cuál es la diferencia entre write y writeline?

Write() imprime el contenido en la misma línea y
Writeline() imprime el contenido dando un salto de línea.

6-¿ Qué es una cadena?

Es una colección de caracteres.

7-¿ Qué es una Variable?

Son pequeñas cajas que existen en memoria de PC.

8-¿ Cómo mostramos el valor de una Variable?

Por medio del write() o el writeline().

9-¿ Cómo le solicitamos un dato al usuario?

Por medio del writeline().

(/ /)

10-¿Cómo se convierte una cadena a un valor numérico?

Por medio de la clase `convert`, utilizando el método necesario dependiendo del tipo final.

11-¿Qué son operadores aritméticos?

Son los que nos permite llevar a cabo operaciones matemáticas.

12-¿De qué forma podemos agrupar operadores aritméticos?

Utilizar los paréntesis para agrupar, ayuda mucho a reducir errores.

Capítulo 3

¿Qué son las estructuras selectivas?

Son aquellas que nos permiten hacer una selección entre dos o varias rutas de ejecución posible.

2-¿Cuándo se ejecuta la sentencia `if`?

Si el resultado de su evaluación es `true`, entonces la sentencia a ejecutar se lleva a cabo. Si el resultado es `false`, entonces la sentencia es ignorada.

3-¿Qué colocamos entre paréntesis en `if`?

Una expresión para evaluar

4-¿Qué es una expresión relacional?

Son las que se utilizan para expresar la relación que existe entre dos valores.

5-¿Cuáles son los tipos de operadores de una expresión relacional?

`True` or `False`

6-¿A qué valores posibles puede evaluar una expresión relacional o lógica?

`True` es usada para indicar que la expresión evaluada es verdadera, y `False` es usada para indicar que la expresión evaluada es falsa.

7- ¿Qué es una expresión lógica?

Son aquellas que pueden tomar uno o dos valores.

8- ¿Cuáles son los operadores de las expresiones lógicas?

11 (or)

1 (not)

8 (and)

9- ¿Qué operador solamente necesita un operando?

La Negación (1)

10- ¿Cómo funciona el switch?

Es necesario colocar entre paréntesis, la variable que utilizaremos para llevar a cabo las comparaciones. Luego, tenemos que crear un bloque de código y colocar dentro del mismo, los casos y el código a ejecutar.

Para indicar un caso, utilizamos Case seguido del valor de comparación y dos puntos.

Existe un caso llamado default, que podemos usar si lo deseamos.

11-¿ Qué es la variable de comparación y como se coloca?

Es la variable utilizada para realizar las comparaciones. Esta se coloca en paréntesis.

12-¿ Cómo definir los casos y como usar el break?

Los casos se definen mediante la palabra case, seguido del valor de comparación mas dos puntos.

El break se utiliza colocandolo donde termina el código de un caso.

Capítulo 4

1-¿ Qué es un ciclo?

Un ciclo es aquello que nos permite repetir algo un número de veces.

2-¿ Cuáles son las partes del ciclo For?

En primer lugar, encontramos la inicialización. En esta sección, le damos a la Variable Control su valor inicial. La segunda sección, lleva una condición en forma de expresión relacional. Luego tenemos el código aquí colocamos la parte del código que queremos que se repita.

Por último, se añade la sección del incremento.

3-¿Cómo colocamos el valor inicial de Contar en un ciclo For?

Declarando una variable e inicializándola desde donde queremos iniciar a contar.

4-¿Cómo colocamos el valor final de Contar en ciclo For?

ponemos una condición que especifica hasta donde contara. Ej: $n \leq 11$.

5-¿Cómo se lleva a cabo el incremento en un ciclo For?

Debemos que modificar el incremento e indicar como se incrementa nuestra variable control.

6-¿Cómo funciona el ciclo do while?

Permite que cierto código se repita mientras una condición se evalúa como verdadera.

7-¿Porque el ciclo while se lleva a cabo al menos una vez?

Este ciclo puede repetir el código un número desconocido de veces.

8. ¿Se necesita punto y coma al finalizar el ciclo do while?

Si.

9. ¿Cómo funciona el ciclo while?

Si la Condición se evalúa como verdadera, entonces se puede ejecutar el código. Después del código, se regresa a la Condición.

10. ¿Cuántas veces se puede repetir el ciclo while?

un número de veces desconocido dependiendo la condición.

11. ¿Qué tipo de condición podemos colocar en el ciclo while?

En la Condición, colocamos una expresión lógica o relacional.

12. ¿Se coloca un bloque de código en ciclo while?

Si

<https://github.com/orlandolara1/ejercicio-1-4cap.git>