Trabajo Practico N ° 1

Paradigmas de Programación

**Ejercicio 1.**

**Ejercicio 2.** indicar cuál de las siguientes afirmaciones referentes a la Programación Declarativa (PD) e Imperativa (PI) es falsa o verdadera. Justique la respuesta.

1) Las instrucciones de un programa declarativo son un conjunto de inferencias lógicas.

VERDADERO: En los Programas declarativos se basan el resultado en base en lo Que quiero que haga y no en el Como quiero que lo haga, en los programas declarativos las instrucciones suelen expresarse mediante premisas y hechos y el programa lo resuelve por medio de deducciones lógicas validas los llamamos Las inferencias lógicas

2) Un programa en PD se corresponde con las instrucciones para resolver un problema.

FALSO: FALSO: En la programación declarativa no se especifica como debo resolver el problema mediante instrucciones. Mas bien como quiero que se resuelva el problema diciendo lo que tiene que hacer y que debe hacer mediante un código fuente, En la Programación Imperativo declaramos instrucciones para que resuelvan el problema paso a paso. En la programación declarativa no se especifica como debo resolver el problema mediante instrucciones. Mas bien como quiero que se resuelva el detallando las argumentos del problema

**Ejercicio 3.**

Distinga entre:

1. Tipos, Ligadura estática de tipos y ligadura dinámica de tipos.

TIPOS: los tipos son datos (Entero, Real, Carácter, String, etc.) o un tipo mas complejos (Arreglos, Clases, etc.) sirven para hacer variables mas correctos clasifican los valores y las expresiones

LIGADURA ESTATICA DE TIPO: Ocurren en tiempo de compilación, donde se hace una comparación de tipos y detecta los errores de tipos incompatible no se puede reemplazar durante la ejecución del programa

LIGADURA DINAMICA DE TIPO: Ocurren en tiempo de ejecución, se compara los tipos durante el tiempo de ejecución cambiando su valor aquellos que pueden ser redefinido en ese momento

2. Coacción, error de tipos y comprobación de tipos.

Coacción: Se refiere al a asignación de tipos de dato en otro, puede ser Explicita (Realizado por el programador) o implícita (realizado por la maquina)

Error de tipos: Se refiere a cuando dos tipos se comparan en tiempo de ejecución o en tiempo de compilación y no son compatibles entonces se produce un error durante la operación

Comprobación de tipos:

3. Compatibilidad de tipos nominal y compatibilidad de tipos estructural.

Compatibilidad Nominal: Una comparación nominal es cuando dos tipos son comparados y tienen el mismo nombre entonces decimos que son compatibles

Compatibilidad Estructural:

**Ejercicio 4.**

En tiempo de ejecución, cuando entramos en un bloque las variables locales ya no contienen los valores que almacenaban la última vez que fue ejecutado dicho bloque. ¿Por qué ocurre esto?

Ocurren por que el tipo de dato es dinámico y la variable cambia de dato de forma implícita (Asignado por el sistema) al terminar si su ejecución en la operación este pude ser desechada para liberar memoria

**Ejercicio 5.**

Dado el siguiente esqueleto de programa escrito en C, dibuja un rectángulo alrededor de cada uno de sus bloques, etiquétalos con las letras A, B, C, etc. y responde a las siguientes preguntas.

int x, y, z;

fun1()

{

int j, k, l;

{

int m, n, x;

...

}

}

fun3()

{

int y, z, k;

...

}

void main()

{ ... }

1. Mencione un bloque que esté anidado dentro de otro bloque.

|  |
| --- |
| {  int j, k, l;  {  int m, n, x;  ...  }  } |

2. ¿En qué ámbitos son accesibles a la vez las variables globales x, y, z?

|  |
| --- |
| int x, y, z; |

3. ¿Qué variables son accesibles desde la función main?

|  |
| --- |
| void main()  { ... } |

4. ¿Son k de fun1 y k de fun3 la misma variable? Razona la respuesta.

No son la misma variable ya que al entrar al bloque de fun1 se le asigna un valor de tipo de dato y al terminar el bloque el dato es eliminado de el almacenamiento

5. ¿Son las variables globales y y z accesibles desde fun3 ?. Justifique la respuesta.

6. ¿En qué bloques j hace referencia a una variable local? ¿En qué bloques j es una variable no local?

7. ¿En que bloque o bloques pueden usarse a la vez m y k?

**Ejercicio 6.**

**Ejercicio 7.**

Indicar cuál de las siguientes afirmaciones es cierta. Justificar su respuesta.

1. Un analizador léxico (scanner) es un programa que divide una secuencia de caracteres (el programa) en una secuencia de componentes sintácticos primitivos: las instrucciones.

VERDADERO:

FALSO :

2. La compatibilidad de tipos, el alcance de las variables y la signatura de las funciones (coincidencia del número de parámetros en una llamada con los parámetros formales) forman parte de la semántica estática del lenguaje.

VERDADERO:

FALSO :

**Ejercicio 8.**

Indicar cuál de las siguientes afirmaciones es cierta. Justificar su respuesta.

1. Una desventaja de los lenguajes de programación declarativos es que, para resolver un mismo problema, se requieren en general más líneas de código que usando otros lenguajes de programación más convencionales, como Python.

FALSO: Los lenguajes declarativos pueden usar solo una línea de código y al resolver el problema en otro lenguaje las soluciones pueden ser más concisa, Python no es un lenguaje de declaración

2. Al ser de mayor nivel, los programas declarativos resultan más fáciles de mantener que los correspondientes programas imperativos.

VERDADERO: Los lenguajes declarativos son mas fáciles de mantener esto se debe a que el problema redacta su solución al Que y no al Como quiero que resuelva el problema esto se debe a es mas descriptivo y se orienta mas a los resultados y además el lenguaje utiliza operaciones de alto nivel lo que reduce el riesgo de errores

3. Los lenguajes declarativos carecen de aplicaciones prácticas.

FALSO:

**Ejercicio 9.**

Indica cuál de las siguientes afirmaciones sobre la programación declarativa en comparación con la programación imperativa es VERDADERA o FALSA:

1. Mayor potencia expresiva. VERDADERO

2. Mantenimiento más simple VERDADERO

3. Menor tamaño del código producido. VERDADERO

4. Establecer el cómo proceder FALSO

**Ejercicio 10.**

Indica cuál de las siguientes afirmaciones es VERDADERA o FALSA.

El lenguaje ensamblador:

1. Más cerca del lenguaje máquina que de los lenguajes de alto nivel VERDADERO

2. Más cerca de los lenguajes de alto nivel que del lenguaje máquina FALSO

3. No es un lenguaje de programación FALSO

**Ejercicio 11.**

Clasifique los siguientes lenguajes para saber si son compilados/intepretados/hibridos.

Lenguajes: Fortran, JavaScript, Mathematica, C,Java, Perl, Pascal, Haskell, Scala, PHP,

Fortran

JavaScript :

Mathematica

C

Java

Perl

Pascal

Haskell

Scala

PHP INTERPRETADO

**Ejercicio 12.**

Indica cuál de las siguientes afirmaciones es VERDADERA o FALSA.

En un lenguaje débilmente tipado:

1. Un valor de un tipo puede ser tratado como de otro tipo VERDADERO

2. Un valor de un tipo nunca puede ser tratado como de otro tipo FALSO

3. Un valor de un tipo puede ser tratado como de otro tipo siempre que se realice una conversión de forma explícita

4. Las anteriores respuestas no son correctas FALSO

**Ejercicio 13.**

Imperativo, declarativo y orientado a objetos son:

1. Modos de compilar el código fuente de un programa de ordenador

2. Modos de definir el pseudocódigo de un programa de ordenador

3. Paradigmas de programación

4. Las anteriores respuestas no son correctas

**Ejercicio 14.**

En base a lo aprendido sobre ligaduras, respondas las siguientes preguntas:

1. ¿Qué es una ligadura? ¿Cómo se pueden clasificar los tiempos de ligadura? ¿A qué características o criterios de evaluación, en general, afecta el tiempo de una ligadura?

2. ¿El concepto de ligadura, está vinculado a la sintaxis o a la semántica de un lenguaje de programación? Justifique y ejemplifique.

3. ¿Cuál es la diferencia entre una ligadura implícita y una explicita? Brindar ejemplos de cada una de ellas.