

## LABORATORIO DE COMPUTACIÓN I – PRACTICA INTEGRADORA

### Parte 1: INTRODUCCIÓN A PROGRAMACION

1. ¿Que es Netbeans?

Es un framework de desarrollo	
Es un entorno de desarrollo o IDE	
Es un servicio de .java para los desarrolladores	

2. ¿Qué es el JDK?

Es un sistema operativo para desarrollo	
Es un Servidor de aplicaciones	
Es un entorno de ejecución y biblioteca de componentes	

3. Indique con un numero el orden que se sigue en el proceso de compilación y ejecución en JAVA

Código precompilado (.class)	
Código Fuente (.java)	
Interpretación Sistema Operativo	

4. ¿Cuál es la signatura de Main?

public void Main()	
public static void main(String [] args)	
private static String Main()	

5. ¿Qué carácter se usa para añadir un comentario de una sola línea?

/	
//	
/* */	

## Parte 2: USO DE VARIABLES

1. ¿Clasifique las siguientes declaraciones en tipo valor o tipo referencia?

double valor = 10	
Integer x=0;	
String nombre = "JUAN";	

2. Según las reglas y recomendaciones en cuanto a nombrado de variables y métodos, marque con una cruz las escritas correctamente.

int while;	
string nombreEmpleado;	
void OrdenaPorApellido()	
void CONVERSOR()	
long main;	

3. ¿Por qué el siguiente código da error?

```
public static void main(String [] args)
{
    Integer bucleCuenta;
    System.out.println("El valor es" + bucleCuenta);
}
```

4. Al finalizar la ejecución de este fragmento de código, ¿qué valor tienen las variables objetoCuenta, prefijoValor y sufijoValor?

```
public static void main(String [] args)
{
    int objetoCuenta = 50;
    int prefijoValor = ++objetoCuenta;
    int sufijoValor = objetoCuenta++;
}
```

5. Indique el tipo de conversión (explícita/implícita) y el valor de la variable que corresponda.

Código	Tipo conversión	Valor de variable
<pre>public static void main(String [] args) {     int intValue= 112;     long longValor=intValue; }</pre>		longValor =
<pre>public static void main(String [] args) {     long longValor = Integer.MaxValue;     int intValue= (int) longValor; }</pre>		intValue =

### Parte 3: INSTRUCCIONES

1. Indique verdadero o falso: Un bloque y su bloque padre pueden tener una variable con distinto nombre.
2. Indique verdadero o falso: Bloques hermanos pueden tener variables con el mismo nombre.
3. Reescriba el siguiente bloque de código usando for para lograr el mismo objetivo:

```
public static void main(String [] args)
{
    int numero = 10;
    while (numero < 0)
    {
        System.out.println("El valor de numero es " + numero);
        numero--;
    }
}
```

## Parte 4: MÉTODOS Y PARÁMETROS

1. Indique que produce el error de compilación, justifique su respuesta.

```
class ExampleReturningValue
{
    public static void main(String [] args)
    {
        int x;
        x = sumarTres( );
        System.out.println("X es igual a" + bucleCuenta);
    }

    private int sumarTres ( )
    {
        int a,b;
        a = 2;
        b = 2;
        c = 2;

        return a + b + c;
    }
}
```

2. Si la llamada a un método fuera de la siguiente forma

```
public static void main(String [] args)
{
    CallMyMethod("sumar", 2.6);
}
```

Como escribiría la estructura del método que procese dicha llamada.

3. Dado el siguiente fragmento de código escriba un nuevo método aplicando sobrecarga de métodos de tal manera que reciba 4 parámetros y los multiplique.

```
class Overloading
{
    static int multiplicarNumeros(int a, int b)
    {
        return a + b;
    }
}
```

// completar método sobrecargado aquí

```
public static void main(String [] args)
{
    System.out.println (multiplicarNumeros (1, 2));
    // llamada al método faltante
    System.out.println (multiplicarNumeros (1, 2, 3, 4));
}
}
```

## Parte 5: Algoritmos

### - String

Cree una aplicación de Consola que permita por parte del usuario el ingreso por teclado de una cadena de texto, posteriormente el sistema deberá permitir el ingreso de una segunda cadena de texto. Finalmente el sistema deberá emitir el mensaje "ENCONTRADO" o "NO ENCONTRADO" según sean el caso de que la segunda cadena existe o no existe en la primera cadena. Valide que la cantidad de caracteres de la segunda cadena sea menor o igual a la cantidad de caracteres de la primera cadena.

Ej.:

Entrada Cadena 1: jhsjdhsjdhsd8787dsds**ds8ds8d**7sd6ds76d

Entrada Cadena 2: **ds8ds8d**

Resultado: "ENCONTRADO"

### - Repetitiva

Diseñar un algoritmo que realice el promedio de N números. La cantidad de números a ingresar debe ser indicada por el usuario, valide que la cantidad de números a ingresar sea mayor a cero.

Los números ingresados pueden ser decimales y serán ingresados por pantalla por el usuario.

### - Recursiva

Diseñar un algoritmo que solicite el ingreso de una cadena de texto y que posteriormente y mediante recursión muestre cada una de las letras de la cadena en forma individual por cada pasada de la recursión. Se recomienda usar **charAt**

### - Arreglo

Codifique un algoritmo que cargue en un array 10 números enteros mayores a 0, todos los números cargados deben ser distintos entre sí, al finalizar la carga, muestre los números cargados.

## Parte 6 OBJETOS

### 1. ¿Qué es una clase?

Una construcción sintáctica con nombre que describe un comportamiento y atributos comunes.	
Es la instancia de un objeto.	
Es un sinónimo de estructura.	

### 2. ¿Qué es un objeto?

Es una clasificación sin instancia.	
Es una instancia de una clase.	
Es una clase abstracta.	

### 3. Escriba la definición de la clase Cliente cuyo alcance debe ser público con los siguientes datos:

Tipos:

apellido, nombre: String.

dni: int.

sueldo: double.

nroCliente: int.

### 4. Cree 2 instancias (objetos) de la clase Cliente, en una clase que posea un método main, haciendo uso de new, y asigne los siguientes valores en c/u de ellas:

Instancia 1:

apellido "Alonso"

nombre: "Eugenia"

dni: 27456987

sueldo: 2698.85

nroCliente: 13654

Instancia 2:

apellido, "Pereira"

nombre: "Victor"

dni: 26987456

sueldo: 3695.45

nroCliente: 12852

5. Muestre el valor asignado en nombre, apellido y de dni de cada uno de los objetos mediante un `System.out.println`
6. En qué se diferencia una clase interface, una clase abstracta y una clase concreta.