

## 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	X
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	X

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

Código precompilado (.class)	2
Código Fuente (.java)	1
Interpretación Sistema Operativo	3

4. ¿Cuál es la signatura de Main?

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

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

/	
//	X
/* */	

## Parte 2: USO DE VARIABLES

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

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

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;	X
void OrdenaPorApellido()	X
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);
}
```

\*Porque la variable bucleCuenta esta declarada pero no inicializada.

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++;
}
```

\*objetoCuenta = 52

\*prefijoValor = 51

\*sufijoValor = 51

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>	Implícita	longValor = 112
<pre>public static void main(String [] args) {     long longValor = Integer.MAX_VALUE;     int intValue= (int) longValor; }</pre>	Explícita	intValue = 9,223,372,036,854 ,775,807

### Parte 3: INSTRUCCIONES

1. Indique verdadero o falso: Un bloque y su bloque padre pueden tener una variable con distinto nombre. **V**
2. Indique verdadero o falso: Bloques hermanos pueden tener variables con el mismo nombre. **V**
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--;
    }
}
* public static void main(String [] args)
{
    for (int numero = 10; numero > 0; numero--)
    {
        System.out.println("El valor de numero es" + 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); ** Variable
    bucleCuenta nunca es declarada**

}

private int sumarTres ( )
{
    int a,b;
    a = 2;
    b = 2;
    c = 2; *variable c es inicializada antes de ser declarada*

    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.

```
public static double CallMyMethod(String operacion, double numero)
{
    return numero;
}
```

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
static int multiplicarNumeros(int a, int b, int c, int d)
{
    return a+b+c+d;
}

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