

## Práctica de la Prueba 1 de Lenguajes de Programación

1. Indique cuál NO es un paradigma de programación visto en clases:

- ☒ a. Future-Oriented Programming
- ☐ b. Object-Oriented Programming
- ☐ c. Functional Programming
- ☐ d. Aspect-Oriented Programming

2. Event-Oriented Programming es usado para escribir:

- ☐ a. Programas que captura los eventos del sistema operativo
- ☐ b. Programas que captura los eventos del usuario
- ☒ c. Programas en base los eventos que puede recibir
- ☐ d. Event-Oriented Programming no existe

3. Los valores de primera clase pueden:

- ☐ a. Ser asignados a una clase.
- ☐ b. Crear clases en tiempo de ejecución.
- ☒ c. Ser asignados a una variable.
- ☐ d. Creados en tiempo de ejecución.

4. Los valores de orden superior pueden:

- ☐ a. Ser asignados a una clase.
- ☒ b. Crear clases en tiempo de ejecución.
- ☐ c. Ser asignados a una variable.
- ☐ d. Creados en tiempo de ejecución.

5. Dado el siguiente código ¿cuáles son los valores de orden superior? (respuesta: una o dos líneas)

```
function foo(lista_de_atributos) {  
    Class Person = new Class(lista_de_atributos);  
  
    return Person;  
}
```

6. Dado el siguiente código ¿Qué sistema de tipo tiene? (respuesta: una o dos líneas)

```
function bar(let x) {  
    let y = 0;  
    for (int i = 0; i < 10; ++i) {  
        y += i;  
    }  
}
```

gradualmente tipado(combinacion)

```

    }
    return value + 2*y;
}

let x = bar(2);
print(value); //da un error por variable sin declarar

```

7. Escriba una función que calcule el producto punto entre dos listas (mirar imagen). Por ejemplo, (calcular-producto-punto '(6 2 5) '(2 0 10)) -> 62.

$$a \cdot b = \sum_{i=1}^n a_i b_i$$

8. Escriba una función (o varias funciones) que determine si un número es perfecto o no, es decir, retorna *verdadero* si el número es perfecto, *falso* en otro caso. Un número perfecto es aquel número que la suma de sus divisores exactos es igual al número, por ejemplo, los divisores de 6 suman 6 (= 3 + 2 + 1).
9. Los lenguajes de programación que son [generales / específicos / Turing-complete / cuánticos] permiten expresar lo máximo que un programa computacional puede realizar.
10. La expresión “x = (function (x, y) {return x + y + 1;}) (10,2)” entrega:
- Función.
  - 13.
  - 10.
  - Un error.