DigitalHouse> Coding School

Certified Tech Developer
The Ultimate Degree

Simulación del parcial de Programación Imperativa

¡Llegó el momento de poner a prueba todo lo que estuvimos viendo a lo largo de estas semanas!

Metodología de evaluación

Se evaluarán los siguientes conceptos sobre el código entregado:

- **FORMA**
  - Que el código esté prolijo e implemente buenas prácticas
  - Que las variables, métodos y funciones tengan nombres descriptivos
  - Que utilices nombres en español o en inglés, pero no ambos
- **LÓGICA**
  - Que la lógica corresponda con lo que solicitan las consignas
  - Que utilices los métodos más adecuados para cada caso
- **FUNCIONAMIENTO**
  - Que el código funcione correctamente, sin arrojar errores
  - Que el código produzca el resultado esperado a partir de los datos suministrados

Formato y entrega

Al terminar el parcial, deben entregar un solo archivo .txt que contenga todos los ejercicios, haciendo uso del formulario que les enviará el docente.

Ejercicio 1:

I. Escribir un objeto literal llamado **empleado** que contenga las siguientes propiedades:
   a. nombre
   b. apellido
   c. edad
   d. salario bruto
   e. ¿está activo?

II. Crear una función denominada ***calcularSalarioNeto*** que retorne el salario neto a partir del salario bruto teniendo en cuenta el descuento porcentual correspondiente a las deducciones impositivas (estos últimos dos datos los recibe por parámetro).

III. Imprimir por consola un mensaje que se componga del nombre, apellido y el sueldo neto del empleado.

Ejercicio 2:

I. Crea un array llamado **programadores**.

II. Crea tres objetos literales con los siguientes datos.
   a. Nombre: Juan, salario bruto: $2750.0, categoría: junior, tareas hechas: 24
   b. Nombre: María, salario bruto: $2750.0, categoría: junior, tareas hechas: 18
   c. Nombre: Lorena, salario bruto: $3900.0, categoría: senior, tareas hechas: 27

III. Crea una función declarada denominada ***aumentarSueldo*** que reciba el array **programadores** como parámetro e incremente en $200 el sueldo bruto. Utilizar una estructura de repetición.

IV. Invocar a la función ***aumentarSueldo*** y mostrar el array **programadores** antes y después de haber sido modificado.

Ejercicio 3:

I. Crear una función de flechas (arrow function) denominada ***calcularPremioPorPerformance*** que reciba un **programador** y que verifica si ha hecho al menos 20 tareas, de ser así, se le paga $10 extra por cada tarea que haya superado el límite mínimo (20). En el caso de que la categoría sea senior, entonces, se le paga $25. La función debe retornar el premio total por performance si corresponde o en su defecto, cero.

II. Agregar una propiedad en cada objeto del array ***programadores*** que sea una constante denominada DEDUCCION_IMPOSITIVA_PORCENTUAL y que tenga un valor de 17.0.

III. Seguidamente, agregar una función expresada denominada ***calcularSalarioNeto*** en cada objeto del array ***programadores***. La misma, debe utilizar el valor de la constante de la propiedad agregada recientemente y la propiedad del sueldo bruto para realizar el cálculo, además, al neto se le sema los premios. Nótese que ya tiene una función declarada que en parte hace esto. Por tal motivo, tiene que reutilizarla.

IV. Crear una función declarada denominada ***mostrarDatosDelProgramador*** que reciba un programador y que muestre por consola un mensaje con los datos de un programador de la siguiente manera:

> "Juan es un/una programador/a junior que tiene un salario bruto de $2750.0 y cobra un salario neto de $2488.5. Hasta el momento, ha hecho 24 tareas."

a. Use las variables para componer el mensaje.
b. Dicha función, se debe meter dentro de un bucle **for** para que se ejecute una vez por cada programador.