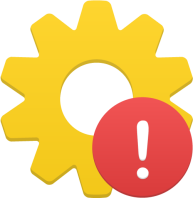
**CFP 8**

**Módulo 2:**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Unidad 2:**

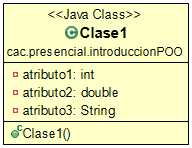
**ENCAPSULAMIENTO**

****

## EJERCICIOS OBLIGATORIOS

### Encapsulamiento, public y private, getters y setters, static, toString()

1. Considerando el siguiente diagrama de clases, la forma correcta de declarar el atributo **atributo1** es:



1. private int atributo1;
2. private int atributo1
3. int private atributo1;
4. private atributo1 int

**Respuesta:** la forma correcta de declarar el atributo1 es la a)

1. Para la clase **Persona** tenemos los atributos: nombre, apellido, edad, sexo. ¿Cuál sería la forma correcta de declararlos?
2. String nombre private;

String apellido private;

int edad private;

String sexo private;

1. private String nombre;

private String apellido;

private int edad;

private String Sexo;

1. private nombre String;

private apellido String;

private edad int;

private sexo String;

**Respuesta:** la forma correcta de declarar los atributos es la c)

1. El método **toString** nos permite mostrar la información completa de un objeto, es decir, el valor de sus atributos.

* Verdadero
* Falso

**Respuesta:** Verdadero

1. Completa los espacios en blanco para crear el método **toString** de la clase Empleado.

**public** String toString(){

String mensaje= "El empleado se llama " + nombre + ", tiene " + edad + " años y un salario de $" + salario + ".";

**return** mensaje;

}

**Respuesta:**no es necesario completar ningún espacio en blanco

1. Arrastra y suelta de las opciones a continuación para definir los métodos set y get.

**public** **class** A {

**private** **int** x;

**public** **int** getX() {

**return** x;

}

**public** **void** setX(**int** x) {

**this**.x = x;

}

}

**Opciones posibles:** x, A, double, int, String, void, return

**Respuesta:** No es necesario agregar ninguna opción adicional

1. ¿Cuáles de los siguientes son modificadores de acceso válidos?
   1. hidden
   2. public
   3. protected
   4. private

**Respuesta:** Los modificadores de acceso valido son public , protected y private.

1. ¿Cuál es la salida del siguiente código?

**public** **class** Vehiculo {

**private** String color;

// Constructor por defecto

**public** Vehiculo() {

}

// Getter

**public** String getColor() {

**return** color;

}

// Setter

**public** **void** setColor(String c) {

color = c;

}

}

**public** **class** MainVehiculo {

**public** **static** **void** main(String[] args) {

Vehiculo v1 = **new** Vehiculo();

v1.setColor("Rojo");

System.***out***.println(v1.getColor());

}

}

**Respuesta:** La salida del siguiente código va a ser la palabra Rojo.

1. Complete el siguiente código para que la variable **CONT** sea compartida por todos los objetos de esa clase.

**public** **class** Contador {

**public** **static** **int** *CONT*= 0;

**public** Contador() {

*CONT*++;

}

}

Respuesta: ya establece la variable CONT como una variable estática compartida por todas las instancias (public static int CONT = 0 )

1. Completa los espacios en blanco para declarar una variable estática.

**public** **static** **int** *x*= 0;

**Respuesta:** No es necesario completar ningún espacio en blanco

1. Arrastra y suelta de las opciones a continuación para crear un código Java válido con encapsulamiento.

**public** **class** Persona {

**private** **int** edad;

**public** **void** setEdad(**int** edad) {

**if** (edad>0)

**this**.edad = edad;

}

}

**Opciones posibles:** int, static, String, public, final, private

**Respuesta:** No es necesario agregar ninguna opción adicional