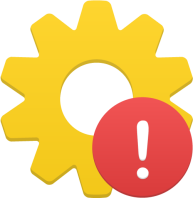
**CFP 8**

**Módulo 2:**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Unidad 4:**

**HERENCIA**

****

## EJERCICIOS OBLIGATORIOS

### Herencia, Subclases y superclases, Sobrecarga, protected vs private, final, Herencia vs Composición, constructores en herencia

1. Completa los espacios en blanco para que la clase Auto herede de la clase Vehiculo.

**public** **class** Auto **extends** Vehiculo {}

1. Completa los espacios en blanco para heredar de la clase Animal e invocar su método en un método main.

**public** **class** Animal {

**public** **void** hacerSonido() {

System.***out***.println("Hola!");

}

}

**public** **class** Perro **extends** Animal {

}

**public** **class** Ej02 {

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

Perro perro = **new** Perro();

perro.hacerSonido();

}

}

1. Métodos **privados** son heredados de la superclase.

* Verdadero
* Falso

1. Dado el siguiente código, determine si la afirmación es verdadera o falsa: “Aquí Auto es la **superclase** y Vehiculo es la **subclase**.”

**public** **class** Auto **extends** Vehiculo {}

* Verdadero
* Falso

1. El modificador de acceso **protected** hace que los miembros sean visibles sólo para las subclases.

* Verdadero
* Falso

1. ¿Cuál es la salida del siguiente código?

**public** **class** Animal {

**protected** **int** patas;

**public** **void** comer() {

System.***out***.println("Animal come");

}

}

**public** **class** Perro **extends** Animal {

**public** Perro() {

patas = 4;

}

}

**public** **class** Ej06 {

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

Perro p = **new** Perro();

p.comer();

}

}

1. Puedes acceder a la superclase desde la subclase utilizando la palabra clave **super**.

* Verdadero
* Falso

1. ¿Cuál es el alcance que deberás asignar a un atributo para que sea accesible desde cualquier método de cualquier objeto de cualquier clase?

* Público
* Privado
* Protegido
* Paquete

1. ¿Cuál es la salida de este código?

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

**private** **void** imprimir() {

System.***out***.println("a");

}

**private** **void** imprimir(String str) {

System.***out***.println("b");

}

**private** **void** imprimir(**int** x) {

System.***out***.println("c");

}

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

A objeto = **new** A();

objeto.imprimir(1);

}

}

1. Completa los espacios en blanco para definir una nueva clase pública Canario, basada en la superclase Pajaro.

**public** **class** Canario **extends** Pajaro {}