**Tema 3:**

1.- Teniendo definido el siguiente código:   
class Persona {   
private String nombre;   
public Persona(String n) {   
this.nombre = n; }   
public void hablar (){System.out.println(“Me llamo “ + this.nombre);   
}   
}   
class Empleado extends Persona {private String puesto;   
public Empleado(String n, String pu) {   
super(n);   
this.puesto = n;   
}   
public void hablar () {System.out.println(“Mi puesto es “ + this.puesto);   
}   
}   
Tras la siguiente ejecución:

public static void main(String[] args) {   
Empleado emp = new Empleado(“Pepe”,”Vendedor”);   
emp.hablar();   
}   
Si el código está bien, ¿qué se muestra en la consola? Si no lo está, ¿cuál es el error?

Seleccione una:

a.?

b.?

2.- **Complete la siguiente afirmación: “Un objeto debe ser manipulado desde el exterior mediante... ”**

Seleccione una:

❏ a. Sus atributos públicos.

❏ b. Sus métodos públicos.

❏ c. Sus métodos estáticos.

❏ d. Sus métodos abstractos

3.- **El método toString de la clase Persona se encarga de mostrar el nombre y el apellido concatenados en mayúsculas. Se sabe que el siguiente código genera una excepción de tipo NullPointerException. Explique con sus palabras cómo lo corregiría (sin usar try – catch).**

public class Persona {   
private String nombre;   
private String apellido;   
public Persona (String nombre, String apellido) {   
nombre = nombre;   
apellido = apellido;   
}   
@Overridepublic String toString() {   
return nombre.toUpperCase() + “ “ + apellido.toUpperCase();   
}   
}   
Respuesta: this.nombre = nombre; igual con apellido

4.- Escribe un programa para: La **Universidad Sin Distancias** (USD) quiere diseñar un sistema de gestión de matrículas de alumnos, del que también forman parte los profesores. Las universidades a nivel nacional se identifican por el nombre, la dirección y el teléfono de información. Desde el punto de vista de la matrícula, la USD se considera como un conjunto de estudiantes, a cada uno de los cuales se le debe asignar un número de identificación personal. Los estudiantes asisten a cierto número de cursos, cada uno de los cuales es impartido por un profesor. En cuanto a los profesores, la USD se divide en departamentos, cada uno de los cuales está formado por varios profesores, uno de los cuales actúa como director del departamento. Cada profesor sólo puede pertenecer a un departamento. A su vez, los departamentos tienen la responsabilidad de impartir uno o más cursos. El sistema debe permitir añadir o borrar estudiantes, departamentos y profesores, así como realizar consultas sobre cada uno de estos estamentos. a. Identifique las clases necesarias para resolver el problema. Indique, para cada una de ellas, sus miembros de clase y su ámbito, así como las relaciones existentes entre estas. Ilustre esto último con un diagrama de clases. b. Suponiendo que existen y están definidos los métodos &quot;get&quot; y &quot;set&quot; para aquellos miembros de clase que haya definido como privados, escriba los métodos: &quot;nuevaMatrícula&quot; que permita realizar la gestión de una nueva matrícula tal y como se ha descripto en la especificación y el método &quot;nuevoProfesor&quot; que permita añadir un nuevo profesor en el organigrama de la Universidad. Indique además en qué clase/es incluiría estos métodos. c) Suponiendo que existen y están definidos los métodos &quot;get&quot; y &quot;set&quot; para aquellos miembros de clase que haya definido como privados, escriba un método &quot;nuevoDepartamento&quot; que permita añadir un nuevo departamento con su correspondiente nuevo conjunto de cursos y profesores. Indique además en qué clase o clases se deberían incluir estos métodos.

d. En el caso de que la Universidad decidiera realizar una división interna en la que los departamentos pertenecieran a una determinada escuela o facultad, ¿qué modificaciones se tendrían que realizar en el diseño de las clases y en el método &quot;nuevoProfesor&quot;?

5.- **Indique cuál de las siguientes afirmaciones es correcta:**

Seleccione una:

❏ a. Las clases que están vinculadas mediante una relación de herencia forman una jerarquía de herencia.   
❏ b. Una subclase es una clase que implementa a otra clase.   
❏ c. Una superclase es una clase que es implementada por otra.   
❏ d. La herencia nos permite heredar pero no reutilizar en un nuevo contexto clases que fueron escritas previamente.