

Asignatura: **Programación II**

Curso: **2018/2019**

Examen:

Final

Fecha: **5-07-2019**

Semestre: **2**

Convocatoria:

Extraordinaria

Ejercicio 1 (4 puntos)

Se desea realizar un programa que guarde una lista de caballos:

- Todos los caballos tienen nombre, edad y sexo.
- Existen dos tipos de caballo: los caballos domésticos y los caballos salvajes.
- Los caballos domésticos, cuando los acaricias agachan la cabeza - **Esto se manifiesta imprimiendo por pantalla “agacho la cabeza”.**
- Los caballos salvajes cuando los acaricias se encabritan. - **Esto se manifiesta imprimiendo por pantalla “me encabrito”.**
- Los caballos domésticos tienen además el nombre del dueño. Los salvajes no tienen dueño.

Se pide realizar un programa en el que se almacene una lista de caballos (pueden ser domésticos o salvajes). El programa debe pedir al usuario, *¿qué tipo de caballo quieres guardar?*, a lo que el usuario contestará *salvaje* o *doméstico*. Además le preguntará el nombre, la edad y el sexo. Si el caballo es doméstico le pedirá además el nombre del dueño.

Cuando el usuario haya introducido todos los caballos se imprimirá por pantalla la lista completa mostrando lo siguiente:

- Caballo llamado: *el nombre*.
- Edad: *la edad*
- Sexo: *el sexo*
- Si me acaricias: *cómo reacciona el caballo*.
- *Soy doméstico y mi dueño se llama / soy salvaje y no tengo dueño.*

Entregar todos los archivos .h y .cpp necesarios. El archivo que contiene la función `main` se debe llamar `main_ejercicio1.cpp`

Rúbrica evaluación

- Insuficiente (25%) - El resultado hace lo que se pide pero sin utilizar herencia.
- Suficiente (50%) - Se ha utilizado herencia, pero las variables/funciones miembro no están en el lugar adecuado.
- Notable (75%) - Se ha utilizado herencia y polimorfismo (si es necesario) de modo adecuado.
- Sobresaliente (100%) - Todos los tipos de datos y la programación utilizada son los más adecuados al problema a resolver, implementando todos los recursos necesarios para una programación limpia y elegante.

Ejercicio 2 (3 puntos)

Realizar una clase **Vector3** que contenga un vector de 3 dimensiones. La clase debe poder reliazar las siguientes funciones a través de sobrecarga de operadores:

- Producto escalar de dos vectores, utilizando el operador *
- Suma, utilizando el operador +
- Resta, utilizando el operador -

Además:

- Debe permitir imprimir el vector por pantalla usando sobrecarga del operador <<

```
cout << elvector << endl; // imprime por pantalla las tres coordenadas
```

- Debe permitir darle los valores al vector, sobrecargar el operador >> de modo que el programa quede:

```
cout << "Introduzca las coordenadas separados por espacios" << endl;  
cin >> elvector;
```

A lo cual es usuario introduciría, por ejemplo *2 1 4 (enter)* y el vector tomaría los valores 2,1,4.

Por ejemplo:

```
cout << "Introduzca las coordenadas separados por espacios" << endl;  
cin >> elvector;  
// el usuario escribe 3 1 5 enter  
cout << elvector << endl; // se imprime por pantalla (3, 1, 5)
```

Realizar un programa que pida por pantalla al usuario 2 vectores y que posteriormente muestre el resultado de su multiplicación, suma y diferencia.

Entregar tres archivos (y sólo estos tres archivos):

- vector3.h
- vector3.cpp
- main_ejercicio2.cpp

Rúbrica evaluación

- Insuficiente (25%) - El resultado hace lo que se pide pero sin usar sobrecarga.
- Suficiente (50%) - Se han sobrecargado al menos 3 operadores y funcionan correctamente
- Notable (75%) - Se han sobrecargado al menos 4 operadores tal y como se pide.
- Sobresaliente (100%) - Todos los tipos de datos y la programación utilizada son los más adecuados al problema a resolver, implementando todos los recursos necesarios para una programación limpia y elegante.

Ejercicio 3 (3 puntos)

Se desea realizar una función **menor()** generica que funcione del siguiente modo:

- El **menor** de dos *números*, será el número de menor valor.
- El **menor** de 2 *vectores de 3 coordenadas (ejercicio 2)* será el vector de menor módulo.
- El **menor** de 2 *string*, será el string con menor número de caracteres.

Se pide implementar un programa que imprima por pantalla:

- El **menor** de los números 3 y 7
- El **menor** de los vectores (2,3,4) y (2,7,9).
- El **menor** de las cadenas de texto “programacion” y “algoritmos”

Para realizar esto, obviamente, debe hacer uso de la función **menor** implementada.

Se reutilizarán los archivos **vector3.h** y **vector3.cpp** del ejercicio2 (¡¡¡no realizar archivos nuevos!!!).
Entregar además un main_ejercicio3.cpp (**ningún otro archivo**)

Rúbrica evaluación

- Insuficiente (25%) - El resultado hace lo que se pide pero sin utilizar una función genérica.
- Suficiente (50%) - Se ha implementado una función genérica y ésta funciona en 2 de los 3 casos.
- Notable (75%) - Se ha implementado una función genérica y ésta funciona en los 3 casos.
- Sobresaliente (100%) - Todos los tipos de datos y la programación utilizada son los más adecuados al problema a resolver, implementando todos los recursos necesarios para una programación limpia y elegante.