

ALUMNO:

EPS

Asignatura: G0460009 Programación II

Curso: 2021/2022
Semestre: 2º

Examen: Final
Convocatoria: Ordinaria

Fecha: 23-05-2022

Parte Práctica (10 puntos; 70% nota final)

Se debe entregar los ficheros de vuestros ejercicios mediante la actividad en el campus antes de la finalización del mismo.

Tiempo: Hasta las 14.00.

Ejercicio 1 [1 punto]

Se debe entregar un único archivo ejercicio1.cpp

Realizar una función que reciba como parámetro un número entero positivo y devuelva el mayor número múltiplo de 3 inferior a dicho número. Realizar un programa que pida al usuario un número y haciendo uso de la función muestre dicho funcionamiento.

Ejercicio 2 [1 punto]

Se debe entregar un único archivo ejercicio2.cpp

Realizar una función que reciba por parámetro una cadena de texto y un número entero positivo n y devuelva la cadena de texto repetida n veces, por ejemplo funcion("hola", 3) devuelve "holaholahola". Se pide realizar un programa que muestre dicho funcionamiento.



Ejercicio 3 [2 puntos]

Se debe entregar un único archivo `ejercicio3.cpp`

1. Implementar una clase que permita almacenar matrícula y marca de un coche.
 - Debe comprobar que la matrícula tiene 4 números y 3 letras, en caso contrario, debe lanzar una excepción.
 - La marca es un texto libre.
2. Sobrecargar los operadores “>” “<” “==” y “!=”
 - El operador “>” indicará si el primer coche es más viejo que el segundo (usando la matrícula)
 - El operador “<” indicará si el primero coche es más nuevo que el segundo (usando la matrícula)
 - El operador “==” indicará si son el mismo coche.
 - El operador “!=” indicará si son coches distintos.
(se presupone que todos los coches son españoles)
3. Sobrecargar el operador “<<” para que muestre por pantalla *marca - matricula*
4. Se pide realizar un programa que muestre dicho funcionamiento.

Ejercicio 4 [6 puntos]

Entregar todos los archivos `.cpp` y `.h` que se consideren oportunos.

Realizar una clase *Persona* que contenga los siguientes atributos (2 puntos):

- nombre
- año de nacimiento.

Implementar:

- Constructor con parámetros para inicializar los atributos (no puede haber constructor sin parámetros). Se debe verificar que la fecha de nacimiento es posterior a 1900, en caso contrario se lanzará una excepción.
- Constructor copia.
- Destructor, que mostrará por pantalla *nombre ha muerto* , siendo *nombre* el nombre de la persona.
- Funciones *getter* y *setter* (las funciones *setter* deben hacer las comprobaciones oportunas)



Realizar una clase *Bombero* que deriva públicamente de *Persona*. Debe contener los siguientes atributos (1 punto)

- Fuegos apagados.
- Longitud del bíceps.

Implementar:

- Constructor con parámetros (no debe contener constructor sin parámetros, y verificación de que los datos son mayores que 0).
- Constructor copia.
- Funciones getter y setter oportunas

Realizar una clase *Futbolista* que deriva públicamente de *Persona*. Debe contener los siguientes atributos (1 punto)

- Goles en su carrera deportiva.
- Millones en el banco.

Implementar:

- Constructor con parámetros (no debe contener constructor sin parámetros, y verificación de que los datos son mayores o iguales que 0).
- Constructor copia.
- Funciones getter y setter oportunas

Realizar un programa que contenga un vector de punteros a Futbolistas y Bomberos y mostrar por pantalla (2 puntos)

- Su nombre y año de nacimiento de todas las personas del vector.
- Si son bomberos: la longitud de su bíceps.
- Si son futbolistas: los millones que tienen en el banco.