

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

Examen 2

(Segundo Semestre 2018)

Indicaciones Generales:

- Duración: (3h.)
- Se podrá usar como material de consulta solo sus apuntes de clase.
- No se pueden emplear variables globales, ni estructuras. No se podrá emplear la clase string. La biblioteca estándar de plantillas (STL) solo se podrá emplear en los puntos donde se indique explícitamente. No se pueden emplear las funciones malloc, realloc, strdup, strtok, tampoco las funciones incluidas en las bibliotecas cstdio, stdio.h o similares. No se podrán definir funciones (ni plantillas) independientes que no estén ligadas como métodos a alguna de las clases planteadas.
- La cláusula friend solo se podrá emplear en el caso de clases auto referenciadas para ligar el nodo con la estructura inmediata a ella.
- Deberá respetar estrictamente el encapsulamiento de datos en las clases a todo nivel, no se considerará en la nota aquellas funciones que violen el encapsulamiento de los datos.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ERICTO DISEÑO DESCENDENTE. Cada módulo NO debe sobrepasar las 20 líneas de código aproximadamente. El archivo main.cpp solo podrá contener la función main de cada proyecto. En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le descontará 0.5 puntos por archivo.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- La presentación, la ortografía y la gramática de los trabajos influirá en la calificación.

Puntaje total: 20 puntos

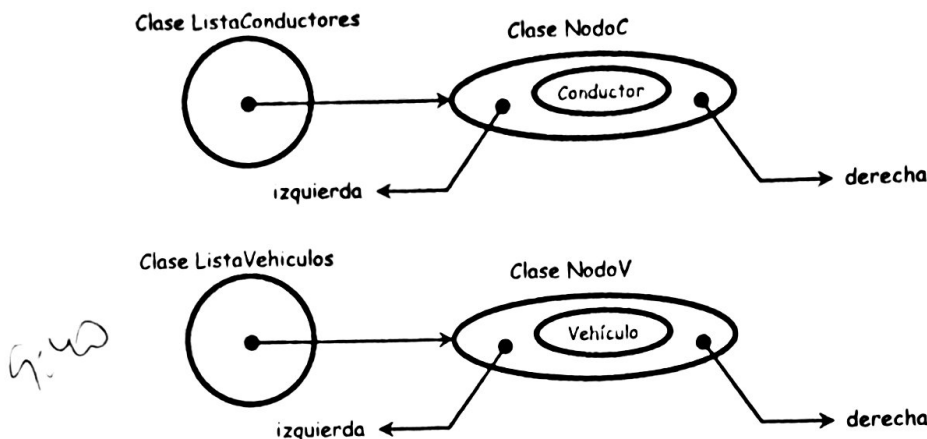
Cuestionario

Se desea elaborar una aplicación orientada a objetos que permita gestionar las infracciones de tránsito cometidas por los diferentes choferes del país. Para realizar esta labor se cuenta con tres archivos CSV, los que se muestran a continuación:

Archivo 1	Archivo 2
63736112,ZAMORA/ZAVALETA/RONAL MANUEL,989320773,K0D-676,T9A-930 45043076,VEGA/VILCARA/CARMELA TERESA,986528258,57E-946,I1D-713,R1T-735	81485316,J5T-691,128 25518120,E1J-798,118
Archivo 3	
103,Detener el vehículo bruscamente sin motivo ,Grave,316 00 201,Dejar mal estacionado el vehículo en lugares permitidos ,Leve,158 00 307,Participar en competencias de velocidad en eventos no autorizados ,Muy Grave,948 00	

En el primer archivo se encuentra la lista de conductores del país, donde aparece el DNI, nombre y teléfono del conductor, seguido por los códigos de las placas de los autos que posee. En el segundo archivo se han registrado las infracciones de tránsito que se han cometido en un periodo de tiempo, donde en cada línea se indica el DNI y la placa del conductor que cometió la falta, seguido del código de la infracción que cometió. En el tercer archivo se tiene un listado de todas las infracciones que se pueden cometer en el país, donde en cada línea aparece el código de la infracción, seguido de su descripción, luego viene la gravedad de la infracción y finalmente la multa por haberla cometido.

La aplicación se desarrollará empleando listas doblemente enlazadas, y para manipularlas se desea cargar los datos de los archivos en objetos cuyas estructuras serán similares a las mostradas en la siguiente figura:



En donde se definen las siguientes clases:

- "Conductor", contiene los datos de un conductor, tendrá como atributos: **dni** (int), **nombre** (char*), **teléfono** (int) y **vehículos** (lista de vehículos que posee - objeto de la clase ListaVehiculos (no un puntero).
- "Vehículo", define el objeto "vehículo". Esta clase estará compuesta por los atributos: **placa** (código de la placa - char*), **infracciones** (códigos de las infracciones cometidas - list <int>).
- "ListaConductores/Clase NodoC", permitirá definir una lista doblemente ligada de conductores.
- "ListaVehiculos/Clase NodoV", permitirá definir una lista doblemente ligada de vehículos.

El atributo del tipo Conductor o Vehículo, del NodoC y NodoV respectivamente, no pueden ser punteros.

PREGUNTA 1 (6 puntos)

Elabore un proyecto denominado "Pregunta01". El proyecto deberá ser capaz de leer los datos del archivo No. 1 y con ellos confeccionar la lista doblemente ligada que contenga los conductores y sus vehículos. Para realizar esto deberá implementar las clases descritas anteriormente. La lista de conductores debe quedar ordenada por el DNI del conductor, mientras que las listas de vehículos por la placa. Las listas de infracciones quedarán vacías.

El proyecto deberá crear un reporte en un archivo de textos en el que se vea los datos de cada conductor correctamente tabulados, etiquetados y con un título que explique la naturaleza de los datos (no podrá emplear el carácter '\t').

PREGUNTA 2 (4 puntos)

Elabore un proyecto denominado "Pregunta02", copie en él las clases desarrolladas en la Pregunta 1 e incorpórela al proyecto, cualquier modificación que haga en este proyecto a la clase incorporada no modificará la calificación de la pregunta 1, NO SE HARÁN EXCEPCIONES.

El proyecto deberá ser capaz de leer los datos del archivo No. 2, y sobre las clases ya definidas deberá agregar los códigos de las infracciones a la lista de infracciones de cada vehículo de cada conductor. Para esto deberá agregar los métodos necesarios a las clases, respetando el encapsulamiento apropiado. El proyecto deberá crear un reporte en un archivo de textos en el que se vea los datos de cada conductor correctamente tabulados, etiquetados y con un título que explique la naturaleza de los datos (no podrá emplear el carácter '\t').

PREGUNTA 3 (5 puntos)

Elabore un proyecto denominado "Pregunta03". El proyecto deberá definir una plantilla de clases que permita definir una lista genérica doblemente ligada ordenada. Esta clase lista tendrá un atributo adicional que hará referencia al último nodo, de manera que permita hacer recorridos secuenciales desde el nodo final hacia el inicial, en el sentido inverso de la lista doblemente ligada.

La plantilla deberá poder crear la lista de forma ordenada según el tipo de dato que se almacene. Asimismo, en la construcción de la lista el atributo adicional mencionado anteriormente debe apuntar al último nodo en todo momento. Adicionalmente, debe incluir un método de impresión que reciba un

atributo entero (0/1) que indique el sentido de impresión a tomar (de inicio a fin o viceversa). Los datos impresos deben estar ordenados, correctamente tabulados, etiquetados y con un título que explique la naturaleza de los datos (no podrá emplear el carácter '\t'). Por otro lado, debe implementar un método de búsqueda que reciba un valor entero a buscar en los objetos almacenados en la lista y retorne un valor de punto flotante.

Para probar la plantilla deberá crear una lista que permita trabajar con objetos de clase Infracción, siendo esta una clase que pueda contener los datos de una infracción (archivo No.3). El método de búsqueda recibirá el código de una infracción y devolverá el valor de la multa. Debe completar la prueba imprimiendo la colección de infracciones usando los dos sentidos de la lista.

PREGUNTA 4 (5 puntos)

Elabore un proyecto denominado "Pregunta04", copie en él las clases desarrolladas en las preguntas 2 y 3 e incorpórelas al proyecto, cualquier modificación que haga en este proyecto a las clases incorporadas no modificará la calificación de las preguntas anteriores, NO SE HARÁN EXCEPCIONES.

Defina y pruebe en el proyecto la clase "EntidadReguladora". Esta clase tendrá como atributos un objeto de clase ListaConductores y un objeto de la clase (plantilla) elaborada en la pregunta 3 que maneje las infracciones. La clase permitirá crear las dos listas, leyendo los datos desde los archivos correspondientes. Finalmente, debe implementar un método que permita leer de un archivo de textos los DNI de conductores e imprimir un reporte de multas acumuladas. Para ello deberá desarrollar un método de búsqueda que obtenga una relación de las infracciones cometidas por todos los vehículos de un conductor (un arreglo de enteros sin reducir repetidos), y así aprovechar luego el método de búsqueda de la clase plantilla implementada en la Pregunta 3 para acumular el monto total por multas.

ENTIDAD REGULADORA DE TRANSITO			
DNI:	Nombre	Teléfono	Monto total por multas
23459784	Juan López Castro	764327721	10787.45
12345678	María Rosa Ruiz Pérez	994567812	3985.99

Los datos deben aparecer correctamente tabulados sin el uso del carácter '\t'.

CONSIDERACIONES FINALES:

- Cree en el computador una carpeta de trabajo con la siguiente ruta: c:\temp\Examen2. En ella colocará los proyectos que den solución al problema planteado.
- De no respetarse el nombre de los proyectos se descontará 1 punto por cada trasgresión.
- La calificación se otorgará por proyecto desarrollado. Por ninguna razón se asignará puntaje a dos o más preguntas por el mismo proyecto.

NO SE HARÁN EXCEPCIONES

- Se prohíbe en esta evaluación el desarrollo de bibliotecas estáticas.

Al finalizar el examen, comprima¹ la carpeta **Examen2** en un archivo con nombre <código del alumno con 8 dígitos>.zip y súbalo a la intranet del curso, en el enlace Documentos, en la carpeta \Examen2\<código del horario>\<aula>. El acceso a la Intranet quedará cerrado automáticamente a las 11:05 am. por lo que el alumno que no suba alguno de los proyectos a la Intranet recibirá como nota CERO en esa pregunta.

NO SE HARÁN EXCEPCIONES.

Profesores del curso: Arturo Oncevay
Miguel Guanira

San Miguel, 11 de diciembre del 2018.

¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión por defecto de Windows (Zip), no use 7z.