

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

10ma práctica (tipo b)

Primer Semestre 2021

Indicaciones Generales:

- Duración: 110 minutos.

Obligatoriamente los alumnos deberán mantener en todo momento el AUDIO Y VIDEO de sus computadores abierto de modo que puedan recibir los comunicados que se hagan durante el examen y la revisión de los trabajos que estén desarrollando. De tener algún problema deberán hacérselo saber de inmediato al profesor de su horario por correo. De no hacerlo, no se aceptarán reclamos alegando que no oyeron las indicaciones.

- No se pueden emplear estructuras, variables globales ni la clase `String`. Tampoco se podrán emplear las funciones `malloc`, `realloc`, `strdup` o `strtok`, igualmente no se puede emplear cualquier función contenida en las bibliotecas `stdio.h`, `cstdio` o similares y que puedan estar también definidas en otras bibliotecas. Tampoco podrá hacer uso de plantillas
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada función NO debe sobrepasar las 20 líneas de código aproximadamente. El archivo `main.cpp` solo podrá contener la función `main` de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. 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.
- NO SE CALIFICARÁN aquellas funciones desarrolladas en el mismo archivo que la función `main`.
- El código comentado NO SE CALIFICARÁ.
- Los proyectos deben obligatoriamente desarrollarse en NetBeans bajo el sistema operativo Windows. No se revisarán los proyectos desarrollados en otros sistemas operativos o IDEs.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.
- Salvo en la sobrecarga de los operadores `>>` y `<<`, no se podrán definir funciones independientes que no estén ligadas como métodos a alguna de las clases planteadas.
- TODOS LOS MÉTODOS Y SOBRECARGAS DE OPERADORES QUE DEFINA DEBEN OBLIGATORIAMENTE SER DESARROLLADOS EN LOS ARCHIVOS .H Y .CPP DONDE SE DECLARE LA CLASE A LA QUE PERTENECE O SU IMPLEMENTACIÓN. DE LO CONTRARIO NO SERÁN EVALUADAS. Se tomará en cuenta en la calificación el uso de comentarios relevantes.

SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO. ESTO Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .cpp O .h CON FECHA U HORA DE CREACIÓN ANTERIOR A LA EVALUACIÓN SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES DADAS EN LA PRUEBA

- Puntaje total: 20 puntos.

Cuestionario:

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 9 del tema: "Clases autoreferenciadas".

Cree una carpeta denominada "Infracciones2021-1Lab10", dentro de ella cree dos carpetas denominadas "Lab10-Parte01" y "Lab10-Parte02", en estas colocará los proyectos solicitados en las preguntas 1 y 2 respectivamente. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL.

PARTE01 (13 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto "LAB10_PREG01" dentro de la carpeta correspondiente, en la cual se declaren las clases descritas con las relaciones necesarias, que permitan manipularlas empleando herencia:

- **Para manejar las infracciones establecidas que se pueden cometer:** la clase se denominará "Infraccion" y deberá contener lo siguiente: 1) un atributo denominado **codigo** con el código de la infracción (int), 2) un atributo denominado **gravedad** definido por una cadena de caracteres (char*) que indicará la gravedad de la falta ("LEVE", "GRAVE" o "MUY GRAVE"), 3) un atributo denominado **multa** (double) que almacenará el pago que debe hacerse por la multa.
- **Para manejar las infracciones cometidas:** La clase se denominará "Falta" y deberá contener lo siguiente: 1) un atributo denominado **licencia** (int) que almacena la licencia del conductor que comete la falta, 2) un atributo denominado **fecha** (int) que almacenará una fecha en el formato AAAAMMDD. 3) un atributo denominado **placa** definido por una cadena de caracteres (char*)
- **Para manejar las faltas leves:** La clase se llamará "Leve" y deberá contener lo siguiente: 1) un atributo denominado **descuento** (double) que almacenará el porcentaje de descuento que se le brinda a los conductores por las faltas leves. Esta clase posee datos heredados de la clase **Infracción**.
- **Para manejar las faltas graves:** La clase se llamará "Grave" y deberá contener lo siguiente: 1) un atributo denominado **descuento** (double) que almacenará el porcentaje de descuento que se le brinda a los conductores por las faltas graves, 2) un atributo denominado **puntos** que almacena la cantidad puntos que pierde un conductor por la falta cometida (int). Esta clase posee datos heredados de la clase **Infracción**.
- **Para manejar las faltas muy graves:** La clase se llamará "MuyGrave" y deberá contener lo siguiente: 1) un atributo denominado **puntos** que almacena la cantidad puntos que pierde un conductor por la falta cometida (int), 2) un atributo denominado **meses** que almacena la cantidad de meses que un conductor será suspendido por la falta cometida (int). Esta clase posee datos heredados de la clase **Infracción**.
- **Para manejar los Nodos:** La clase se denominará "NFalta" y deberá contener lo siguiente: 1) un atributo denominado **pfalta**, este atributo es un puntero de la clase **Infracción**, 2) un atributo denominado **dfalta** perteneciente a la clase **Falta**, 3) un atributo denominado **sig**, este atributo es un puntero a la clase **NFalta** (autoreferenciado).
- **Para manejar la lista:** La clase se denominará "LFalta" y deberá contener lo siguiente: 1) un atributo denominado **lini**, este atributo es un puntero de clase **NFalta**, 2) un atributo denominado **lfin**, este atributo es un puntero de clase **NFalta**.

"DEBE EMPLEAR OBLIGATORIAMENTE LOS NOMBRES DE LAS CLASES Y SUS ATRIBUTOS"

Con las clases indicadas debe realizar las siguientes operaciones:

- En la clase **LFalta** debe implementar el método **leer**, que se encarga de la lectura del archivo "RegistroDeFaltas1.csv" y cargar la información de cada fila en los nodos de la clase **NFalta**, formando una lista ordenada por la fecha, de acuerdo con la infracción cometida. Para este proceso el puntero **lini** debe apuntar al inicio de la lista, y el puntero **lfin** debe apuntar al final de la lista.
- En el desarrollo del método **leer** debe cargar la clase **Falta**, con el resto de los datos del archivo.
- De acuerdo con el código de la falta registrada (si empieza con 2 es leve, con 1 es grave o si empieza con 3 es muy grave) carga la clase correspondiente a la falta, utilizando el puntero **pfalta** de la clase **NFalta**, el tipo de falta **no** se colocará en la estructura, debido a que las Infracciones deberán ser manipuladas con polimorfismo. Los datos correspondientes a las clases derivadas y la clase base Infracción deben leerse del archivo "Infracciones.csv", el cual puede leerse varias veces.
- Finalmente, en la clase **LFalta** debe implementar el método **imprimir**, que se encargue de realizar la impresión de un archivo de prueba debidamente tabulado (**sin usar el carácter '\t'**), que muestre la

fecha, la licencia, la placa del auto y los datos correspondiente a cada infracción cometida. En el desarrollo de este método debe utilizar un método polimórfico, para la impresión de los datos de la infracción.

Consideraciones:

Se le recomienda no usar operadores sobrecargados para el desarrollo de la pregunta anterior, así como revisar al detalle los archivos csv, indicados. Debido a que se leen y almacenan todas las faltas registradas, es posible que un conductor se repita varias veces. Para el desarrollo de la segunda pregunta debe considerar el siguiente código:

```
#include "LFalta.h"

using namespace std;

int main(int argc, char** argv) {
    LFalta lf1, lf2;

    lf1.leer("RegistroDeFaltas1.csv");
    lf2.leer("RegistroDeFaltas2.csv");
    lf1.unir(lf2);
    lf1.imprimir("reporte.txt");

    return 0;
}
```

No puede
cambiar este
código

PARTE 2 (7 puntos): Prueba final.

Desarrolle en la carpeta "Lab10-Parte02" un proyecto denominado "LAB10_PREG02" en el cual se utilizará obligatoriamente las clases desarrolladas en la pregunta anterior. El proyecto ejecutará las tareas descritas a continuación:

- Cargar las listas lf1 y lf2 de acuerdo con lo indicado en la pregunta anterior.
- Desarrollar el método **unir**, el cual se encarga de fusionar la lista lf1 y la lista lf2, considerando una complejidad máxima $O(1)$, la lista resultante debe ser representada en la lista lf1.
- Desarrollar el método **imprimir** que mejorando el reporte de prueba de la pregunta anterior imprima el siguiente reporte:

Fecha	Licencia	Placa	Infracción	Monto	Descuento	Puntos	Meses Susp
2019/02/20	66849703	J2F-853	Muy Grave	948.00	0.00	600	1
2019/02/21	88854166	H3E-885	Muy Grave	474.00	0.00	500	0
2019/02/21	73357516	E6R-293	Grave	316.00	0.25	100	0
2019/02/23	58992942	Q6B-243	Grave	316.00	0.25	100	0
2019/02/23	15559308	T8F-368	Grave	316.00	0.25	200	0
2019/02/24	54447051	G3Y-654	Muy Grave	474.00	0.00	500	0
2019/02/25	87282766	S1D-093	Grave	316.00	0.25	100	0
2019/02/25	43454884	G0F-080	Leve	158.00	0.50	0	0
2019/02/26	21621497	Z7C-785	Leve	158.00	0.50	0	0
2019/02/26	34248357	U0T-459	Grave	316.00	0.25	100	0
2019/02/28	54447051	D5W-773	Muy Grave	474.00	0.00	500	0

Se recomienda revisar los archivos que servirán para la lectura de datos.

Al finalizar la práctica, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para esta esté laboratorio.

Profesores del curso: Miguel Guanira
Rony Cueva

San Miguel, 9 de julio del 2021.