PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1
8va 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 muestres 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.**
  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 7del tema: "Herencia".

Cree una **carpeta** denominada "Infracciones2021-1Lab08", dentro de ella cree dos **carpetas** denominadas "Lab08-Parte01" y "Lab08-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 (15 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto **"LAB08_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 los conductores:** La clase se denominará "**Conductor**" y deberá contener lo siguiente: 1) un atributo denominado **licencia**, este atributo debe almacenar la licencia del conductor (int), 2) un atributo denominado **nombre** definido por una cadena de caracteres (char*).

➢ **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 fecha (int) que almacenará una fecha en el formato AAAAMMDD. 2) 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 las clases **Infracción** y **Falta**.

➢ **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 las clases **Infracción** y **Falta**.

➢ **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 las clases **Infracción** y **Falta**.

➢ **Para manejar los registros de infracciones cometidas**: La clase se denominará "**Registro**" y deberá contener lo siguiente: 1) un atributo denominado lleve, este atributo es de la clase Leve, 2) un atributo denominado lgrave, este atributo es de la clase **Grave,** 3) un atributo denominado lmgrave, este atributo es de la clase **MuyGrave**. Cada objeto solo tendrá cargada la información correspondiente a la falta cometida por el conductor. Por ejemplo, si un conductor tiene una falta grave solo estará cargada la información correspondiente al atributo lgrave. Además, esta clase posee datos heredados de la clase **Conductor.**

➢ **Para manejar los procesos del programa**: La clase se denominará "**Procesa**" y deberá contener lo siguiente: 1) un atributo denominado lregistro, este atributo es un arreglo estático de la clase **Registro**, 2) un atributo denominado cantidad, este atributo almacena la cantidad de elementos de la clase **Registro** que tiene el arreglo (int).

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

Con las clases indicas debe realizar las siguientes operaciones:

- En la clase **Procesa** debe implementar el método **lee,** que se encarga de la lectura del archivo "RegistroDeFaltas.csv" y cargar la información de cada fila en el arreglo lregistro.
- Al leer la licencia del conductor debe cargar la información correspondiente a la clase **Conductor**, utilizando el archivo "Conductores.csv", el cual puede ser leído varias veces.
- 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 en la clase **Registro.**
- Al cargar la falta correspondiente, también debe completar los datos vinculados a la clase **Falta** e **Infracción**, para llenar esta última clase puede leer las veces que sea necesario el archivo "Infracciones.csv".

- Finalmente, en la clase **Procesa** implementar el método **imprime,** que se encargue de realizar la impresión de un archivo de prueba debidamente tabulado (**sin usar el carácter '\t'**), que muestre la licencia, la fecha y la placa del auto, correspondiente a cada falta cometida.

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 ambas preguntas debe considerar el siguiente código:

```
#include "Procesa.h"

int main(int argc, char** argv) {
   Procesa pro;

   pro.lee();
   pro.imprime();
   pro.consolida();
   return 0;
}
```

No puede cambiar este código

PARTE 2(5 puntos): Prueba final.
Desarrolle en la carpeta "Lab08-Parte02" un proyecto denominado **"LAB08_PREG02"** en el cual se utilizará obligatoriamente las clases desarrolladas en la pregunta anterior. El proyecto ejecutará las tareas descritas a continuación:

- Cargar el arreglo lregistro de acuerdo con lo indicado a la pregunta anterior.
- Desarrollar el método **consolida** en la clase **Procesa** que lea el archivo "sancionados.csv" y de acuerdo con las licencias de este archivo, imprima el siguiente reporte:

```
Conductor   : LEYVA SAM DIANA DELIA
Licencia No.: 50200823
----------------------------------------------------------------
Monto de Multas:   3634
Puntos perdidos:   2200
Meses sancionados:    0


Conductor   : AVILA DEL CASTILLO ANNIE
Licencia No.: 54447051
----------------------------------------------------------------
Monto de Multas:   8295
Puntos perdidos:   3500
Meses sancionados:    6

.........
```

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 este laboratorio.

Profesores del curso: Miguel Guanira
Rony Cueva

San Miguel, 25 de junio del 2021.