

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

5ta práctica (tipo b)
(Primer Semestre 2020)

Indicaciones Generales:

- Duración: 110 minutos.

Obligatoriamente los alumnos deberán mantener en todo momento el audio de sus computadores abierto de modo que puedan recibir los comunicados que se hagan durante el examen. 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 variables globales, estructuras, 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.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO 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 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**.
- La cláusula **friend** solo se podrá emplear en el caso de clases autoreferenciadas para ligar el nodo con la clase **inmediata** que encapsula la lista, **en ningún caso adicional**. No se considerará en la nota las clase que violen esto
- Deberá mantener en todo momento el encapsulamiento de todos los atributos de las clases así como guardar los estándares en la definición y uso de todas las clases desarrolladas.
- Salvo en la sobrecarga de los operadores >> y <<, **no se podrán definir funciones (ni plantillas de funciones) independientes que no estén ligadas como métodos a alguna de las clases planteadas.**
- El código comentado **NO SE CALIFICARÁ**.
- 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%.
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 AL EXAMEN 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

Cree una carpeta con el nombre "2020_1_LAB05" y en él desarrolle los proyectos que a continuación se solicitan.

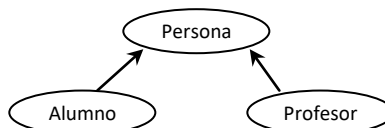
Cuestionario

Se desea simular el sistema de atención de reservas de espacios académicos dentro de la universidad. Se sabe que estas solicitudes se guardan en archivos CSV según se van realizando. Para realizar un proceso de asignación equitativo, se determinó brindarles los espacios o aulas, hasta una determinada capacidad para los alumnos y las de mayor capacidad a los profesores. Por tal motivo se debe generar un programa que utilizando plantillas, listas ligadas y STL de solución a este requerimiento.

PREGUNTA 1 (4 puntos)

Dentro de la universidad existen 2 tipos de personas, los cuales pueden ser alumnos y profesores, ambos pueden acceder a los servicios de reserva de espacios académicos. Como ambos tipos poseen

datos comunes, pero también datos particulares a su clase, se requiere desarrollar una jerarquía de clases como se muestra a continuación:



Los datos miembros o atributos para estas clases serán los siguientes:

Persona: nombre de la persona, código y distrito de residencia.

Alumno: escala de pago, especialidad y facultad a la que pertenece.

Profesor: categoría (CON, AUX, ASO o PRI), dedicación (TC, TP o TPA), sección a la que pertenece y grado académico.

Los tipos de datos asociados a los atributos se indican a continuación (no podrá modificarlos):

Se pide que desarrolle un proyecto denominado **"Pregunta01"** en donde se declaren las tres clases descritas con las relaciones necesaria que permitan manipularlas, utilizando herencia.

| Persona | Alumno | Profesor |
|---------------------------------------------------|----------------------------------------------------|---------------------------------------------------------------------|
| char * nombre; int codigo; char * distrito; | int escala char *especialidad char *facultad | char *categoria char *dedicacion char *seccion char *grado |

La implementación contemplará la definición de atributos, constructores por defecto, constructores copia, destructores, métodos selectores, un método que permita leer los datos desde un archivo de texto, otro que permita imprimirlo en otro archivo de texto. Además las sobrecargas necesarias para el manejo de plantillas. La memoria asignada a las cadenas debe ser óptima.

La función main deberá probar de manera muy sencilla la implementación de estas clases y métodos.

PREGUNTA 2 (5 puntos)

Elabore un proyecto denominado **"Pregunta02"**, y defina una plantilla de clases (class PCola) que permita definir una lista ligada que se comporte como una cola. La plantilla deberá permitir encolar y desencolar alumnos o profesores que reservan espacios académicos. Además permitirá mostrar, en un archivo de textos, los datos contenidos en la cola (desde el inicio hasta el final, sin retirar los datos de la cola). La cola debe manejarse mediante dos punteros, uno al inicio y otro al final de la cola. La impresión de los datos debe estar correctamente tabulada. Para probar su plantilla puede utilizar cualquier dato que le simplifique esta operación. **De no colocar una especificación para la plantilla se considerará al momento de la calificación que el proyecto NO COMPILA.**

PREGUNTA 3 (5 puntos)

Elabore un proyecto denominado **"Pregunta03"** en donde debe definir la clase **Aula**, la cual cuenta con los siguientes atributos:

| Aula |
|-------------------------------------------------------------------------|
| char *clave int capacidad int codigo char *nombre char tipo |

En esta clase los atributos representan la siguiente información: **clave:** guarda el código del espacio académico, **capacidad:** es la cantidad de personas que pueden usar el espacio, **codigo:** representa el código de la persona que lo reservó, **nombre:** el representa el nombre de la persona que lo reservó y **tipo:** representa quien realizó la reserva de la sala un alumno o profesor. Debe implementar los métodos constructores, destructores y selectores necesarios. Así como las sobrecargas adecuadas para su manejo, considerando las sobrecargas de lectura (revise el archivo "Aulas.csv") e impresión.

Una vez que este implementada la clase Aula, utilizando STL desarrolle un **vector** vAula, que utilice la clase mencionada. En la función main debe probar la lectura utilizando el archivo **"Aulas.csv"**, este archivo tiene solos 2 datos, el primero es la clave del aula y el segundo dato es la capacidad del aula. Si

en el proceso de lectura algunos atributos no son cargados, utilice valores por defecto para su llenado (ver reporte, pagina 4). Luego de la lectura pruebe la escritura en un archivo de texto.

PREGUNTA 4 (6 puntos)

Elabore un proyecto denominado **"Pregunta04"** y copie en él las clases desarrolladas en las preguntas anteriores e incorpórelas al proyecto. Además este proyecto debe contener una clase **Reserva** que tiene como atributo un STL vector **vAula**, así como una cola que utilice la plantilla desarrollada para almacenar a los alumnos y una cola que utilice la misma plantilla para almacenar a los profesores, que han requerido reservas de espacios académicos (aulas).

La clase **Reserva** debe tener un método **"carga"** que reciba como parámetros dos cadenas de caracteres, donde se ingresaran los nombres de los archivos a leer. Además este método debe realizar la lectura del archivo **"Aulas.csv"** y almacene los datos en el vector **vAula**, así mismo debe ser capaz de leer el archivo denominado **"Miembros.csv"** que contiene a todas las personas que realizaron una reserva de los espacios académicos, según se describe a continuación:

| |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A,Castro/David,61,Jesús María,1,Ing.Informatica,EEGGCC A,Velarde/Mario,54,Lince,1,Ing.Informatica,FACI P,Caceres/Jose,36,San Juan,ASO,TC,Informatica,Magister |
| El archivo contiene lo siguiente: Tipo Persona (A: Alumno, P: Profesor), nombre, código del miembro, distrito y a continuación los datos del alumno o profesor. Si es el alumno será: escala, especialidad y facultad. Si es el profesor será: categoría, dedicación, sección, grado. |

De acuerdo al tipo de persona se debe cargar la cola de alumnos o la cola de profesores. El orden de reserva es el mismo que tiene el archivo.

Una vez cargadas las tres estructuras, debe desarrollar el método **"asigna"** que recibe como parámetro un número entero que presenta la capacidad máxima que puede tener un aula para ser asignada a un alumno, una cantidad mayor a este entero debe ser asignado a un docente. Este método debe asignar una persona (alumno o docente) en cada aula de acuerdo a su capacidad en el vector **vAula**, registrando los atributos código, nombre y tipo. Para el proceso de asignación de un aula, debe desencolar a la persona de acuerdo al tipo que le corresponda, por ejemplo si la capacidad máxima que puede utilizar un alumno es 4, y el vector presenta un aula de capacidad 4 o menos, debe desencolar un elemento de la cola de alumnos, si en caso el aula que presenta el vector tiene una capacidad mayor a 4, debe desencolar el elemento de la cola de profesores. Si no hay más elementos en la cola requerida **no debe asignar** dicho espacio o sala.

Finalmente debe implementar el método **"imprime"** que permita mostrar en un archivo de texto, la asignación de aulas a los diferentes solicitantes sean alumnos o profesores. Si en caso la sala no se asignó a ninguna persona, debe mostrar en el reporte la palabra **"Sin Asignar"**. A continuación se muestra el reporte de reservas considerando como parámetro una capacidad de 4 personas.

| Reservas Asignadas |
|---------------------------------------------------------------------------------------------------------|
| Aula: A01 Capacidad: 4 Codigo: 92419 Nombre: LOZADA/YINO/MARTHA Tipo: A ----- |
| Aula: A02 Capacidad: 4 Codigo: 67560 Nombre: CAMACHO/AGUERO/ELIA ROSA Tipo: A ----- |
| Aula: A03 Capacidad: 6 Codigo: 2506 Nombre: SANTISTEBAN/MEZA/ROSA Tipo: P ----- |

Aula: B08
Capacidad: 8
Codigo: 0
Nombre: Sin Asignar
Tipo: S

No puede usarse ningún atributo adicional a lo indicado. Desarrollar los métodos o sobrecargas que necesite.

Anotaciones finales

Al finalizar el laboratorio, comprima¹ la carpeta "2020_1_LAB05" y súbalo a Paideia.

Profesores del curso: Rony Cueva
Miguel Guanira E.

San Miguel, 17 de julio del 2020

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