

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**LENGUAJES DE PROGRAMACIÓN 1**

**4ta 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

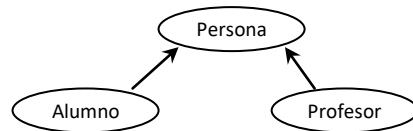
**Cree una carpeta con el nombre "LABORATORIO4-2020-1" y en él desarrolle los proyectos que a continuación se solicitan.**

**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 realizan las solicitudes, pero al ser asignadas, se brinda mayor prioridad a los profesores según su dedicación. Por tal motivo se debe generar un programa que utilizando polimorfismo y listas ligadas de solución a este requerimiento.

**PREGUNTA 1** (10 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 empleando polimorfismo.

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

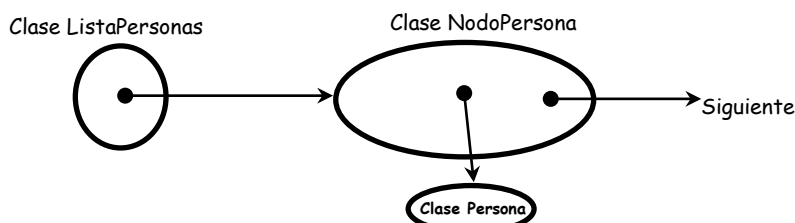
La implementación contemplará la definición de atributos, constructores por defecto, destructores, métodos selectores, un método que permita leer los datos desde un archivo de texto, otro que permita imprimirlo en otro archivo de texto y otro que devuelva la prioridad de reserva que tiene cada persona (1: TC, 2:TP 3:TPA y 4:Alumno). La memoria asignada a las cadenas debe ser óptima.

La función main deberá probar de manera muy sencilla la implementación de estas clases y su manejo empleando polimorfismo (lectura, impresión y prioridad).

## PREGUNTA 2 (10 puntos)

Elabore un proyecto denominado "Pregunta02", copie en él las clases definidas en la pregunta 1 e incorpórelas al proyecto, luego desarrolle en él las clases que permitan definir y manipular la lista ligada de la figura 1.

Figura 1



La lista debe quedar ordenada de acuerdo a la prioridad de las personas que solicitan una reserva, primero los de prioridad 1, luego la prioridad 2 y así respectivamente. La prioridad no es un atributo de ninguna clase. El proyecto debe ser capaz de leer la información del 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.

El tipo de persona **no se colocará** en la estructura, debido a que las personas deberán ser manipuladas con polimorfismo. Luego del llenado de la lista se deberá emitir un reporte, con el orden en que se asignarán las reservas de los espacios académicos, generando un archivo de texto similar al que se muestra a continuación:

Reservas Asignadas	
Cantidad de TC:	11
Cantidad de TP:	6
Cantidad de TPA:	13
Cantidad de Alumnos:	35
-----	
Codigo:1354	
Nombre:PISCOYA/BRAVO/CESAR BRUNO	
Dedicacion:TC	
Seccion:Economia	
-----	
Codigo:6113	
Nombre:SHIRAKAWA/MIRANDA/MADELEINE	
Dedicacion:TC	
Seccion:Derecho penal	
-----	
...	
-----	
Codigo:2506	
Nombre:SANTISTEBAN/MEZA/ROSA	
Dedicacion:TP	
Seccion:Derecho civil	
-----	
Codigo:2160	
Nombre:MALDONADO/RAMIREZ/SILVANA	
Dedicacion:TP	
Seccion:Pintura	
-----	
Codigo:1677	
Nombre:ARROYO/GORDILLO/MARIA HELI	
Dedicacion:TP	
Seccion:Derecho penal	
-----	
.....	

El reporte debe considerar las cantidades de alumnos y profesores que han solicitado reservas de acuerdo a la prioridad. Luego de esta cabecera debe mostrarse el listado ordenado de reservas asignadas. No puede usarse ningún atributo para guardar la cantidad de tipos de reservas consideradas. Para generar la información mostrada puede omitir algunos datos del método imprimir de la pregunta 1 y desarrollar los métodos o sobrecargas que necesite.

### Anotaciones finales

Al finalizar el laboratorio, comprima<sup>1</sup> la carpeta "**LAB03-2020-1**" y súbalo a Paideia.

Profesores del curso: Rony Cueva  
Miguel Guanira E.

San Miguel, 10 de julio del 2020

<sup>1</sup> 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.