

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

**LENGUAJES DE PROGRAMACIÓN 1**

**7ma práctica (tipo b)**  
**Primer Semestre 2022**

**Indicaciones Generales:**

Duración: 110 minutos.

**SOLO ESTÁ PERMITIDO EL USO DE APUNTES DE CLASE. NO PUEDE UTILIZAR FOTOCOPIAS NI MATERIAL IMPRESO, TAMPOCO PODRÁ EMPLEAR HOJAS SUELTAS.**

- No se pueden emplear **variables globales, estructuras, ni objetos** (con excepción de los elementos de iostream, iomanip y fstream). **No puede utilizar la clase (o el tipo de datos) 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. **NO PODRÁ EMPLEAR PLANTILLAS EN ESTE LABORATORIO**
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ERICTO 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 el archivo main.cpp deberá colocar un comentario en el que coloque claramente su nombre y código, **de no hacerlo se le descontará 0.5 puntos en la nota final**.
- El código comentado NO SE CALIFICARÁ. De igual manera NO SE CALIFICARÁ el código de una función si esta función no es llamada en ninguna parte del proyecto o su llamado está comentado.
- 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%.
- 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.**

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

- **Puntaje total: 20 puntos.**

**INDICACIONES INICIALES**

Cree un proyecto de C++ en NetBeans siguiendo estrictamente las indicaciones que a continuación se detallan:

- La unidad de trabajo será **t:\** (Si lo coloca en otra unidad, no se calificará su laboratorio y se le asignará como nota cero)
- Cree allí una carpeta con el nombre **"CO\_PA\_PN\_Lab07\_2022\_2"** donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento se le descontará 3 puntos de la nota final). **Allí colocará los proyectos solicitados en la prueba.**

**Cuestionario:**

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en los capítulos 6 y 7 del tema: "Programación orientada a objetos" y "Operadores Sobrecargados".

**PARTE01 (10 puntos): CREACIÓN DE LAS CLASES**

Se solicita que desarrolle un proyecto **"LAB07\_PREG01\_CLASES"** dentro de la carpeta correspondiente, **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL**, en la cual se definan clases para el manejo de los clientes y pedidos realizados. Así como una serie de operadores sobrecargados que permitan manejar estas clases. A continuación, se definen las clases que serán necesarias:

- **Para manejar los medicos:** La clase se denominará "**medico**" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**int**), 2) un atributo denominado **nombre** definido por una cadena dinámica (char \*) de caracteres, 3) un atributo denominado **tarifa** (**double**),
- **Para manejar los pacientes:** La clase se denominará "**paciente**" y deberá contener lo siguiente: 1) un atributo denominado **dni** (**int**), 2) un atributo denominado **nombre** definido por una cadena dinámica (char \*) de caracteres, 3) un atributo denominado **lcitas**, definido por un arreglo de la clase **citas**, considerar un máximo de 100 elementos, 4) un atributo denominado **telefono** (**int**), 5) un campo denominado **totalgastado** (**double**) que llevará la suma total de lo gastado por un paciente en citas médicas.
- **Para manejar las citas:** La clase se denominará "**cita**" y deberá contener lo siguiente: 1) un atributo denominado **dni** (**int**) que representa el dni del paciente, 2) un atributo denominado **codigo** (**int**) que representa el código del médico, 3) un atributo denominado **fecha** (**int**) que representa la fecha donde se realiza el pedido con el formato **aaaammdd**, 4) un campo denominado **tarifa** (**double**) que almacenará el monto gastado por la cita o consulta, inicialmente en 0.

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

Las operaciones que se permitirá realizar a través de sobrecargas de operadores se definen a continuación:

#### • **Lectura:**

- Sobrecargando el operador >> de modo que permita leer **un** médico de un archivo csv. La operación (**arch>>medico;**) involucrará un archivo y un objeto de la clase "**medico**". Una línea de archivo tendrá la siguiente forma:

506117,EDGAR MUNOZ VERGEL,337.03  
(codigo, nombre del medico, tarifa)

- Sobrecargando el operador >> de modo que permita leer **un** paciente de un archivo csv. La operación (**arch>>paciente;**) involucrará un archivo y un objeto de la clase "**paciente**". Una línea de archivo tendrá la siguiente forma:

43704548,Vizcardo/Maribel,960851431  
(dni, nombre del paciente, telefono)

- Sobrecargando el operador >> de modo que permita leer **una** cita de un archivo csv. La operación (**arch>>cita;**) involucrará un archivo y un objeto de la clase "**cita**". Una línea de archivo tendrá la siguiente forma:

36471787,600493,16/01/2023  
(dni,código del médico, fecha)

Estas sobrecargas pueden ser definidas como parte del archivo de la clase que manejan.

#### • **Agregación y totalización:**

- Sobrecargando el operador += de modo que permita agregar **una** cita a **un** paciente. La operación (**paciente += cita;**) colocará los datos de una cita (**cita**) al final de su lista de citas de un paciente. No debe modificar los atributos que no están involucrados en la operación.
- Sobrecargando el operador <=. La operación (**cita <= arreglomedico;**) que involucrará a una clase **cita** y un arreglo del tipo **medico**, buscará en el arreglo, el código del médico contenido en el atributo **codigo** y si lo encuentra le asignará su tarifa al atributo **tarifa** de la clase **cita**, de lo contrario no modificará la variable.
- Sobrecargando el operador ++ de modo que se pueda calcular el total gastado de un paciente. La operación (**paciente++**) sumará todas las tarifas contenidas en las citas de un paciente (**paciente**) y la asignará al atributo **totalgastado** del paciente. No debe modificar los atributos que no están involucrados en la operación.

#### • **Impresión:**

- Sobrecargando el operador << de modo que permita imprimir la información de **un** paciente. La operación (**arch << paciente;**) permitirá imprimir en un archivo de textos los datos contenidos en un objeto de la clase "**Paciente**". El formato será el siguiente:

DNI	NOMBRE	TELEFONO
43704548	Vizcardo/Maribel	883451172
RELACION DE CITAS		
No.	FECHA	CODIGO DEL MEDICO
1)	02/03/2022	534710
2)	...	...
MONTO TOTAL GASTADO: 4567.89		
=====		

El reporte debe estar perfectamente tabulado (**sin usar el carácter '\t'**).

### Consideraciones:

La solución debe contemplar la elaboración del proyecto de implementación, y la prueba de las sobrecargas en el main, no hay problema si para esta labor se excede en el número de líneas. Las pruebas de las sobrecargas deben ser realizadas lo más simple posible, pero que se muestre claramente que son correctas (se puede emplear "hard code").

### PARTE 2(10 puntos): Prueba final.

Desarrolle un proyecto denominado "**LAB07\_PREG02\_CARGA**" en el cual se utilizará obligatoriamente las clases desarrolladas en la pregunta anterior. El proyecto ejecutará las tareas descritas a continuación utilizando las sobrecargas definidas anteriormente:

- Leer los datos de los médicos contenidos en el archivo "medicos.csv" y colóquelos en un arreglo fijo de objetos de la clase **medico**. Se deben utilizar las sobrecargas desarrolladas en la primera pregunta.
- Leer los datos de los pacientes contenidos en el archivo "pacientes.csv" y colóquelos en un arreglo fijo de objetos de la clase **paciente**. Se deben utilizar las sobrecargas desarrolladas en la primera pregunta.
- Con los dos arreglos anteriores cargados correctamente, leer los datos de las citas o consultas contenidos en el archivo "citas.csv" y colóquelos en un arreglo fijo **lcitas** de cada **paciente**. Se deben utilizar las sobrecargas desarrolladas en la primera pregunta.
- Utilizando las sobrecargas de la primera parte actualizar las tarifas de las citas del arreglo de la clase **paciente**. Al procesar cada paciente emplear la sobrecarga correspondiente para actualizar el atributo **totalgastado**.
- Emitir un reporte en el que se muestren los datos de todos los pacientes y sus respectivas citas. Se debe utilizar la sobrecarga de impresión de la primera pregunta.

Para realizar las operaciones solicitadas debe desarrollar las funciones necesarias, en una biblioteca exclusiva para estas acciones. **No puede manejar variables enteras o arreglos en el main para el control del número de pacientes o medicos.** Finalmente, no puede agregar más atributos de los indicados en el enunciado, pero si los métodos que sean necesarios en cualquiera de las clases.

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 este laboratorio.

Profesores del curso: Miguel Guanira  
Rony Cueva

San Miguel, 4 de noviembre del 2022.