

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

3ra práctica (tipo b)
Primer Semestre 2023

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, 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 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 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_Lab03_2023_1"** 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 el capítulo 4 del curso: "Funciones y alcance de variables". En este laboratorio se desarrollará una **biblioteca estática de funciones** en la que se implementen sobrecargas de operadores y funciones que le permitan solucionar el problema planteado. Además, se le solicitará enlazar una biblioteca estática, proporcionada por los profesores del curso para que sea utilizada obligatoriamente en los proyectos solicitados.

En la carpeta solicitada anteriormente, cree **dos carpetas** denominadas **"Parte1_CrearBiblioteca"** y **"Parte2_UsarBiblioteca"**, en estas últimas colocará los proyectos solicitados en las partes 1 y 2 respectivamente de este laboratorio. **DE NO COLOCAR ALGUNO DE ESTOS REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL.**

PARA LA APERTURA DE LOS ARCHIVOS DE TEXTOS SE LE ESTÁ PROPORCIONANDO UNA BIBLIOTECA ESTÁTICA, OBLIGATORIAMENTE DEBERÁ ENLAZAR Y UTILIZAR ESTA

BIBLIOTECA EN LOS PROYECTOS QUE ASÍ LO REQUIERAN. DE UTILIZAR OTRAS FUNCIONES PARA ESTE FIN SE LE DESCONTARÁ 2 PUNTOS EN SU NOTA FINAL.

TAMBIÉN SE LE PROPORCIONARÁ UN ARCHIVO DE CABECERA (.h) CON LAS ESTRUCTURAS DE DATOS QUE UTILIZARÁ PARA DESARROLLAR ESTE LABORATORIO, NO PUEDE MODIFICAR ESTE ARCHIVO, SI MODIFICA DE ALGUNA FORMA ESTE ARCHIVO SE LE DESCONTARÁ 5 PUNTOS EN SU NOTA FINAL.

PARTE 01 (12 puntos): CREACIÓN DE LA BIBLIOTECA ESTÁTICA

Se solicita que desarrolle una biblioteca estática en la cual se defina una serie de operadores sobrecargados que permitirá manejar las estructuras de datos que se le está proporcionando.

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

➤ **Lectura:**

- Sobrecargando el operador `>>` de modo que permita leer un curso de un archivo de textos. La operación (`arch >> curso;`) involucrará un archivo de textos y una variable de tipo `"StCurso"`. La sobrecarga deberá devolver un valor de tipo `bool`, `true` si se pudo leer la información y `false` si se llegó al final de archivo. Una línea de archivo tendrá la siguiente forma:

```
INF263 Algoritmia 3.75 35030611 INGA_FLORES_CESAR_ADOLFO
(código y nombre del curso, créditos, código y nombre del profesor)
```

El arreglo alumnos queda vacío, y los valores para los campos `numeroDeAlumnos` e `ingresos` deben quedar en cero.

- Sobrecargando el operador `>>` de modo que permita leer un alumno de un archivo de textos. La operación (`arch >> alumno;`) involucrará un archivo de textos y una variable de tipo `"StAlumno"`. La sobrecarga deberá devolver un valor de tipo `bool`, `true` si se pudo leer la información y `false` si se llegó al final de archivo. Una línea de archivo tendrá la siguiente forma:

```
20211,1397 ZEVALLOS/PRADO/RICARDO S 50 G2
(semestre de ingreso, código, nombre, modalidad, porcentaje y esacla)
La modalidad puede ser Virtual (V), Semi presencial (S) o presencial (vacío)
El porcentaje solo aparece si es semi presencial
```

El arreglo cursos queda vacío, y los valores para los campos `numeroDeCursos` y `costoTotal` deben quedar en cero. Tener en cuenta que dos alumnos pueden tener el mismo código, pero sus semestres de ingreso serán diferentes.

- Sobrecargando el operador `>>` de modo que permita leer un registro de matrícula de un archivo de textos. La operación (`arch >> registroMat`) involucrará un archivo de textos y una variable de tipo `"StRegistroDeMatricula"`. La sobrecarga deberá devolver un valor de tipo `bool`, `true` si se pudo leer la información y `false` si se llegó al final de archivo. Una línea de archivo tendrá la siguiente forma:

```
INF263 20231,8320
(código del curso, semestre del alumno, código del alumno)
```

➤ **Agregación:**

- Sobrecargando el operador `+=` de modo que permita agregar un curso a un arreglo de cursos. La operación (`arregloCursos += curso;`) colocará los datos de un curso (`struct StCurso`) al final de un arreglo de tipo `struct StCurso`. Como no se puede llevar la cuenta de datos en el arreglo, el arreglo deberá tener como último dato un curso con código `"XXXXXX"`.
- Sobrecargando el operador `+=` de modo que permita agregar un alumno a un arreglo de alumnos. La operación (`arregloAlumnos += alumno;`) colocará los datos de un curso (`struct StAlumno`) al final de un arreglo de tipo `struct StAlumno`. Como no se puede llevar la cuenta de datos en el arreglo, el arreglo deberá tener como último dato un alumno con semestre cero.
- Sobrecargando el operador `*=` de modo que permita agregar un alumno al curso en que se matriculó. La operación (`arregloCursos *= registroMat;`) buscará el curso contenido en la variable `registroMat`, de tipo `StRegistroDeMatricula` en el arreglo `arregloCursos` y agregará al alumno a la lista de matriculados, modificando el número de alumnos.

- Sobrecargando el operador *= de modo que permita agregar un curso al alumno en el que se matriculó. La operación (**arregloAlumnos *= registroMat;**) buscará el alumno contenido en la variable **registroMat**, de tipo **StRegistroDeMatricula** en el arreglo **arregloAlumnos** y agregará el curso a la lista de cursos matriculados, modificando el número de cursos.

➤ **Impresión:**

- Sobrecargando el operador << de modo que permita imprimir la información de un curso. La operación (**arch << curso;**) permitirá imprimir en un archivo de textos los datos contenidos en una variable de tipo **"struct StCurso"**. El formato será el siguiente:

CODIGO	NOMBRE	CREDITOS
INF263	Algoritmia	3.75
PROFESOR: INGA FLORES CESAR ADOLFO [35030611]		
Alumnos matriculados:		
Semestre	Código	
20221	4576	
20232	1052	
...	...	
Numero de alumnos: ...		
Total ingresado: ...		

- Sobrecargando el operador << de modo que permita imprimir la información de un alumno. La operación (**arch << alumno;**) permitirá imprimir en un archivo de textos los datos contenidos en una variable de tipo **"struct StAlumno"**. El formato será el siguiente:

SEMESTRE	CODIGO	NOMBRE	MODALIDAD	ESCALA
20211	1397	ZEVALLOS/PRADO/RICARDO	SEMIPRESENCIAL	CON 50% 2
Códigos de cursos matriculados:				
	INF111			
	MEC123			
	...			
Numero de Cursos: ...				
Costo total: ...				

Los reportes deben estar perfectamente tabulado (**sin usar el carácter '\t'**).

Consideraciones:

La solución debe contemplar la elaboración de: 1) un proyecto de implementación y prueba de las sobrecargas, denominado **"Biblioteca Fuentes_2023_1"**, 2) un proyecto que genere la biblioteca estática (.a) denominado **"Biblioteca Compilada_2023_1"** y 3) un proyecto donde se pruebe la biblioteca ya compilada **"Prueba Biblioteca Compilada_2023_1"**. La prueba de las sobrecargas para el primer y tercer proyecto deben ser hecha lo más simple posible pero que muestre claramente que son correctas (**No se debe solucionar aquí el problema de la parte 2, de hacerlo no se le considerará esta parte en la nota**). Los tres proyectos deberán colocarse en la carpeta **"Parte1_CreaBiblioteca"**.

EN LA BIBLIOTECA ESTÁTICA, NO PUEDE DESARROLLAR OTRAS FUNCIONES O SOBRECARGAS ADICIONALES A LAS SOLICITADAS.

PARTE 2 (8 puntos): REUTILIZACIÓN DE LA BIBLIOTECA ESTÁTICA.

Desarrolle un proyecto denominado **"UsoDeBibEstaticaEnAplicacion"** (dentro de la carpeta **"Parte2_CreaBiblioteca"**) en el cual se utilizarán obligatoriamente las sobrecargas de la biblioteca estática (compilada) **"libBiblioteca_Compilada_2023_1.a"**.

LA PREGUNTA NO SE EVALUARÁ SI EN ESTE PROYECTO NO SE USAN (O NO SE VE QUE USAN) LAS SOBRECARGAS IMPLEMENTADAS EN LA BIBLIOTECA COMPILADA (.A) Y EN LA BIBLIOTECA Y ESTRUCTURAS PROPORCIONADAS.

El proyecto ejecutará las tareas descritas a continuación utilizando las sobrecargas definidas en la biblioteca:

- Leer los datos de los cursos contenidos en un archivo de textos como se muestra a continuación y los coloque en un arreglo de estructuras del tipo **"struct StCurso"**:

INF263	Algoritmia	3.75	35030611	INGA_FLORES_CESAR_ADOLFO
MEC270	Procesos_De_Manufactura	4	83265244	PAIRAZAMAN_ALAMO_MOISES_MIGUEL
...				

- b) Leer los datos de los alumnos contenidos en un archivo de textos como se muestra a continuación y los coloque en un arreglo de estructuras del tipo **“struct StAlumno”**:

20211,1260	GORDILLO/CASTRO/RONAL			G2
20211,1397	ZEVALLOS/PRADO/RICARDO	S	50	G2
...		

- c) Leer los datos de cada matrícula contenidos en un archivo de textos como se muestra a continuación:

INF246	20212,9719
MEC270	20211,9958
...	...

La información de una línea será colocada en una variable del tipo **“struct StRegistroDeMatricula”**, y con ella debe completar la información de los cursos y alumnos matriculados de los arreglos.

- d) Calcular los totales de ingresos y costos de alumnos y cursos empleando la información de las escalas de pagos del archivo “Escalas.txt”.
- e) Emitir un reporte con la información completa de cursos y alumnos. El reporte debe tener un título y subtítulos adecuados.

Al finalizar la práctica, comprima la carpeta dada en las indicaciones iniciales 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.

Profesores del curso: Rony Cueva
Erasmus Gómez
Miguel Guanira

San Miguel, 28 de abril del 2023.