

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

LENGUAJES DE PROGRAMACIÓN 1

1ra práctica (tipo b)

Segundo 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`, `omanip` y `fstream`). **No puede utilizar la clase `string`**. Tampoco se podrán emplear las funciones `malloc`, `realloc` o `strdup`, 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_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 el capítulo 1 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 **"CrearBiblioteca_Parte1"** y **"UsarBiblioteca_Parte2"**, allí 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.** **NO SE HARÁN EXCEPCIONES**

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 O INSTRUCCIONES 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 cliente de un archivo de CSV. La operación (`arch >> cliente;`) involucrará un archivo de textos y una variable de tipo `"struct Cliente"`. 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:

```
79464412,PORTUGAL RAFFO ALEXANDER,3902394
(DNI, nombre y teléfono del cliente)
El resto de campos deben inicializarse adecuadamente
```

- Sobrecargando el operador `>>` de modo que permita leer un producto de un archivo de CSV. La operación (`arch >> producto;`) involucrará un archivo de textos y una variable de tipo `"struct Producto"`. 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:

```
BIT-434,Campana Extractora modelo Glass,375.09,10
(código, descripción, precio y stock del producto)
El resto de campos deben inicializarse adecuadamente
```

- Sobrecargando el operador `>>` de modo que permita leer un pedido de un archivo de CSV. La operación (`arch >> pedido`) involucrará un archivo de textos y una variable de tipo `"struct Pedido"`. 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:

```
JXD-139,50375303
(código del producto pedido y DNI del cliente que hace el pedido)
El campo restante debe inicializarse adecuadamente.
```

➤ Agregación:

- Sobrecargando el operador `+=` de modo que permita agregar un pedido a un arreglo de clientes. La operación (`arregloClientes += pedido;`) tomará el DNI del `"pedido"` (`struct Pedido`), buscará el cliente en el arreglo `"arregloClientes"` y agregará el código y precio del pedido al final del campo `productosEntregados` (arreglo), incrementando el campo `cantidadProductosEntregados` y el campo `montoTotal`.

Como no se puede llevar la cuenta de datos en el arreglo, el arreglo deberá tener como último dato un DNI con valor cero.

- Sobrecargando el operador `+=` de modo que permita agregar un pedido a un arreglo de productos. La operación (`arregloProductos += pedido;`) tomará el código del producto del `"pedido"` (`struct Pedido`), buscará el producto en el arreglo `"arregloProductos"` y agregará el DNI del cliente al final del campo `"clientesServidos"` (arreglo) si el producto cuenta con stock o al campo `"clientesNoServidos"` (arreglo) si el producto NO cuenta con stock, modificando las cantidades correspondientes. La operación **debe** colocar el precio del producto en el campo `precioProducto` del pedido. La sobrecarga deberá devolver un valor de tipo `bool`, `true` si el producto tenía stock y `false` si no lo tenía. De ser el caso debe descontar el stock. El stock NO puede ser negativo.

Como no se puede llevar la cuenta de datos en el arreglo, el arreglo deberá tener como último dato un código con valor `"XXXXXXX"`.

➤ **Impresión:**

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

54761731	LOYOLA IBEROS ANGEL GUSTAVO	951334343	11389.27	Productos entregados: VZV-827	VZV-827	(*)
----------	-----------------------------	-----------	----------	-------------------------------	---------	-----

DNI, nombre, teléfono, monto total, y lista de productos entregados
(*) Si no se le entregaron productos debe colocar el mensaje "NO SE LE ENTREGARON PRODUCTOS".

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

SJD-818 Refrigeradora CoolStyle	281A Steel	3387.75	15
Cilentes atendidos:	54761731	94326707	(*)
Cilentes no atendidos:	71984468	46462527	22280452 (**)

Código, descripción, precio y stock final.
(*) Si no atendieron pedidos debe colocar el mensaje: "NO SE ATENDIERON PEDIDOS"
(**) Si se atendieron todos pedidos debe colocar el mensaje: "NO HAY CLIENTES SIN ATENDER"

En ambos casos, los datos deben aparecer con un formato adecuado y alineados correctamente, tanto para los enteros (hacia la derecha), valores de punto flotante (a la derecha y al punto decimal) y cadenas de caracteres (a la izquierda). **No se dará puntaje a esta parte si no respeta esta condición. NO SE HARÁN EXCEPCIONES.**

Consideraciones:

La solución debe contemplar la elaboración de: 1) un proyecto de implementación y prueba de las sobrecargas, denominado **"Fuentes_Biblioteca_2023_2"**, 2) un proyecto que genere la biblioteca estática (.a) denominado **"Biblioteca_Compilada_2023_2"** y 3) un proyecto donde se pruebe la biblioteca ya compilada **"Prueba_Biblioteca_Compilada_2023_2"**. 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. Los tres proyectos deberán colocarse en la carpeta **"CrearBiblioteca_Parte1"**.

NO SE DEBE SOLUCIONAR AQUÍ EL PROBLEMA DE LA PARTE 2, DE HACERLO NO SE LE CALIFICARÁ LA SEGUNDA PARTE DE ESTE LABORATORIO.

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 **"BibEstaticaEnUnaAplicacion"** (dentro de la carpeta **"UsarBiblioteca_Parte2"**) en el cual se utilizarán obligatoriamente las sobrecargas de la biblioteca estática (compilada) **"libBiblioteca_Compilada_2023_2.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:

- a) Leer los datos de los clientes contenidos en un archivo CSV como se muestra a continuación y los coloque en un arreglo de estructuras del tipo **"struct Cliente"**:

50365593,SUAREZ BRAVO MARIELLA TATIANA,965004473
95751208,CORONEL ZEGARRA FIORELLA,925608637
...

- b) Leer los datos de los productos contenidos en un archivo CSV como se muestra a continuación y los coloque en un arreglo de estructuras del tipo **"struct Producto"**:

YOT-530,Deshumecedor DM-190H,941.73,13
NSR-955,Cocina a gas Blogna,2591.44,14
...

c) Leer los datos de cada pedido contenidos en un archivo de CSV como se muestra a continuación:

PVZ-181,26290971
ICX-503,27912250
...

La información de cada línea deberá completar los pedidos de ambos arreglos (clientes y pedidos).
DEBE EMPLEAR OBLIGATORIAMENTE LOS OPERADORES SOBRECARGADOS.

d) Emitir un reporte con la información completa de los productos y los clientes. 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, 1 de septiembre del 2023.