

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

PROGRAMACIÓN 2
1ra práctica (tipo b)
Segundo Semestre 2024

Indicaciones Generales:

- Duración: 110 minutos.

NO SE PERMITE EL USO DE APUNTES DE CLASE, FOTOCOPIAS NI MATERIAL IMPRESO

- 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`, `memset`, `strtok` 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 **"Lab01_2024_2_CO_PA_PN"** 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 (14 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. Estas estructuras de datos están relacionadas con los pedidos que se realizan a un restaurante de la ciudad.

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 los datos de un plato de un restaurante de un archivo de tipo CSV. La operación (`arch >> plato;`) involucrará una variable de archivo y una variable de tipo `"struct Plato"`. La sobrecarga no deberá devolver valor alguno. Una línea de archivo tendrá la siguiente forma:

```
BR-611,1/2 POLLO - PAPAS - ENSALADA,46.90,BRASA
(código, nombre, precio y tipo de un plato)
El resto de campos deben inicializarse adecuadamente
```

- Sobrecargando el operador `>>` de modo que permita leer los datos de un repartidor de un archivo de tipo CSV. La operación (`arch >> repartidor;`) involucrará una variable de archivo y una variable de tipo `"struct Repartidor"`. La sobrecarga no deberá devolver valor alguno. Una línea de archivo tendrá la siguiente forma:

```
BCJ176,Bravo Cortes Janet Flor,Bicicleta
(código, nombre y tipo de vehículo que emplea del repartidor)
El resto de campos deben inicializarse adecuadamente
```

- Sobrecargando el operador `>>` de modo que permita leer los datos de un pedido de un archivo de tipo TXT. La operación (`arch >> pedido;`) involucrará una variable de archivo y una variable de tipo `"struct Pedido"`. La sobrecarga no deberá devolver valor alguno. Una línea de archivo tendrá la siguiente forma:

```
12484697 AD-546 2 LAF361 2.53
(dni del cliente, código del plato solicitado, cantidad pedida,
código del repartidor y la distancia que recorrerá en la entrega)
El resto de campos deben inicializarse adecuadamente
```

➤ **Operaciones:**

- Sobrecargando el operador `<=` de modo que permita determinar el precio de un plato. La operación: `pedido <= arregloPlatos;` tomará el código del plato pedido de la variable `"pedido"` (`struct Pedido`), buscará el plato en el arreglo `"arregloPlatos"` (`struct Plato`). Si lo encontró, asignar el precio del plato al pedido, incrementar los totales del plato y devolver el valor `true`, si no encontró devolverá un valor `false`. Como no se puede trabajar con la cantidad de datos en el arreglo, el arreglo deberá tener como último dato un registro con el código con valor "FIN".
- Sobrecargando el operador `<=` de modo que permita agregar un pedido a un repartidor. La operación `arregloRepartidores <= pedido;` agregará el pedido (`struct Pedido`) al repartidor correspondiente (`struct Repartidor`). El pedido realizado por un cliente puede estar compuesto por varios platos, pero cada plato aparecerá en el archivo en un registro (línea) independiente, no necesariamente de manera consecutiva en el archivo, por esta razón, la sobrecarga deberá buscar el repartidor asignado y agrupar los platos por el cliente que hizo el pedido, por lo que un repartidor no puede tener clientes repetidos en su lista de órdenes de compra. Tampoco el pedido de un cliente no puede tener platos repetidos por lo que tendrá que acumularlos. Como no se puede trabajar con la

cantidad de datos en el arreglo, el arreglo deberá tener como último dato un registro con el código con valor "FIN".

- Sobrecargando el operador "!" de modo se permita calcular el monto que debe pagar un cliente por el total de platos pedidos y el pago por el envío. La **operación !ordenDeCompra** deberá poder calcular estos montos en los campos correspondientes de la **"ordenDeCompra"** (**struct OrdenDeCompra**). El pago por envío estará dado en función a la distancia a recorrer, si la distancia es menor de 8 km la tarifa será 10.50, si está entre 8 y 12 km la tarifa será 14.80, si está entre 12 y 20 km la tarifa será 23.60 y mayores que 20 la tarifa será de 31.70.
- Sobrecargando el operador "!" de modo se permita calcular el monto que recibirá el repartidos por todas sus entregas. La **operación !repartidor** deberá poder calcular estos montos en los campos correspondientes del **"repartidor"** (**struct Repartidor**).

➤ Impresión:

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

BR-611	1/2 POLLO - PAPAS - ENSALADA	46.90	23	1078.70
--------	------------------------------	-------	----	---------

Código, nombre, precio, total de platos y monto recaudado

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.**

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

LAF361	Lozada Arguedas Felicita Valentina	Motocicleta	225.89	(*)
ORDENES ENTREGADAS:				
15290194	2.45	138.60	8.00	(**)
Platos solicitados:				
- BR-419	37.90	3	113.70	(***)
- SA-330	24.90	1	24.90	
22679759	15.80	135.50	23.60	
...				

(*) Código, nombre tipo de vehiculo, monto total por envios del repartidor

(**) DNI, distancia, monto por la orden pedida y pago por envio.

(***) Código, precio, cantidad y total del plato pedido.

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_2024_2"**, la biblioteca tendrá el mismo nombre que el proyecto,
- 2) un proyecto que genere la biblioteca estática (.a) denominado **"Biblioteca_Compilada_2024_2"** y
- 3) un proyecto donde se pruebe la biblioteca ya compilada **"Prueba_Biblioteca_Compilada_2024_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 **"Parte1_CrearBiblioteca"**.

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

EN LAS PRUEBAS PARA LOS PROYECTOS 1 Y 3 SE PUEDE HACER DIRECTAMENTE EN LA FUNCIÓN MAIN, SE PUEDE HACER USO DE "HARD CODE" Y SE PUEDE SOBREPASAR LA CANTIDAD EL NUMERO DE LÍNEAS, SOLO EN ESTA PARTE DE LA PRUEBA.

EN LA BIBLIOTECA ESTÁTICA, NO PUEDE DESARROLLAR OTRAS FUNCIONES O SOBRECARGAS ADICIONALES A MENOS QUE ESTÉN DIRECTAMENTE RELACIONADAS A LAS SOBRECARGAS SOLICITADAS.

PARTE 2 (6 puntos): REUTILIZACIÓN DE LA BIBLIOTECA ESTÁTICA.

Desarrolle un proyecto denominado "UnaAplicacionConBibEstatica" (dentro de la carpeta "Parte2_UsarBiblioteca") en el cual se utilizarán obligatoriamente las sobrecargas de la biblioteca estática (compilada) "libBiblioteca_Compilada_2024_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 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 platos contenidos en un archivo CSV como se muestra a continuación y los coloque en un arreglo de estructuras del tipo "struct Plato":

EN-910,DELICIA MEDIANA CON POLLO,22.90,ENSALADAS
BR-136,1/2 POLLO BBQ/HOT - ANTICUCHO,60.50,BRASA
...

código, nombre, precio y tipo de un plato

- b) 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 Repartidor":

PRT150,Pairazaman Raffo Tatiana Delicia,Bicicleta
SYI157,Sam Yallico Ines,Motocicleta
...

DNI del cliente, Nombre del Cliente

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

20371464	BR-823	3	NCJ980
22598563	BE-265	1	JNV387
...			

dni del cliente, código del plato solicitado, cantidad pedida, código del repartidor y la distancia que recorrerá en la entrega

Con la información de cada línea deberá modificar los arreglos. DEBE EMPLEAR DE MANERA OBLIGATORIA LOS OPERADORES SOBRECARGADOS.

- d) Emitir un reporte con la información completa de los platos y los repartidores. 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
Andrés Melgar
Erick Huiza
Miguel Guanira

San Miguel, 6 de septiembre del 2024.