

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

6ta 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**. **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_Lab06_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 descontarán 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 (14 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto **"LAB06_PREG01_CLASES"** dentro de la carpeta correspondiente, **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁN 3 PUNTOS DE LA NOTA FINAL**, en la cual se definirán clases para el manejo de los datos de un almacén, sus los clientes y sus respectivos pedidos históricamente obtenidos. Así como una serie de operadores sobrecargados que permitan manejar estas clases. A continuación, se definen las clases que serán necesarias:

- **Para manejar los productos entregados:** La clase se denominará **"ProductoEntregado"** y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**char ***) 2) un atributo denominado **precio** (**double**).

- **Para manejar solo los clientes:** La clase se denominará "**Cliente**" y deberá contener lo siguiente: 1) un atributo denominado **dni** (**int**) 2) un atributo denominado **nombre** (**char ***), 3) un atributo denominado **telefono** (**int**), 4) un atributo denominado **productos_entregados**, definido por un arreglo estático de la clase **ProductoEntregado**., 5) un atributo denominado **cantidad_productos_entregados** (**int**), 6) un atributo denominado **monto_total** (**double**).
- **Para manejar los productos:** La clase se denominará "**Producto**" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**char ***), 2) un atributo denominado **descripcion** (**char ***), 3) un atributo denominado **precio** (**double**), 4) un atributo denominado **stock** (**int**), 5) un atributo denominado **clientes_servidos** definido por un arreglo estático (**int**), 6) un atributo denominado **clientes_no_servidos** definido por un arreglo estático (**int**), 7) un atributo denominado **cantidad_clientes_servidos** (**int**), 8) un atributo denominado **cantidad_clientes_no_servidos** (**int**).
- **Para manejar los pedidos:** La clase se denominará "**Pedido**" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**char ***), 2) un atributo denominado **dni_cliente** (**int**), 3) un atributo denominado **precio_producto** (**double**).
- **Para manejar el almacén:** La clase se denominará "**Almacen**" y deberá contener lo siguiente: 1) un atributo denominado **arreglo_clientes**, definido por un arreglo estático de la clase **Cliente**, 2) un atributo denominado **cantidad_clientes** (**int**), 3) un atributo denominado **arreglo_productos**, definido por un arreglo estático de la clase **Producto**, 4) un atributo denominado **cantidad_productos** (**int**)

"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 cliente de un archivo de CSV. La operación (**arch >> cliente;**) involucra un archivo de csv y una variable de tipo "**class 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 csv y una variable de tipo "**class 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 csv y una variable de tipo "**class 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.
```

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

- **Agregación:**

- Sobrecargando el operador **+=** de modo que permita agregar **un** pedido a **un** cliente. Para usar la operación (**cliente += pedido;**) deberá agregar el código y precio del pedido al final del campo **productos_entregados** (arreglo de **ProductoEntregado**), incrementando el campo **cantidad_productos_entregados** y el campo **monto_total**.

- Sobrecargando el operador += de modo que permita agregar un pedido a un producto. Para usa la operación (**producto += pedido;**), agregar el DNI del cliente al final del campo "**clientes_servidos**" (arreglo) si el producto cuenta con stock o al campo "**clientes_no_servidos**" (arreglo) si el producto NO cuenta con stock, modificando las cantidades correspondientes. La operación **debe** colocar el precio del producto en el campo **precio_producto** 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.

- **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 "**class 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 "**class Producto**". El formato será el siguiente:

```
SJD-818 Refrigeradora CoolStyle 281A Steel 3387.75 15
Clientes atendidos: 54761731 94326707 (*)
Clientes 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 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.

PARTE 2(6 puntos): Prueba final.

Desarrolle un proyecto denominado "**LAB06_PREG02_ALMACEN**" 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 para popular los datos de un almacén:

- Leer los datos de los clientes contenidos en un archivo CSV como se muestra a continuación y los coloque en **arreglo_clientes**, atributo de la clase Almacen:

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

- Leer los datos de los productos contenidos en un archivo CSV como se muestra a continuación y los coloque en **arreglo_productos**, atributo de la clase Almacen:

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

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

```
FVZ-181,26290971
ICX-503,27912250
...
```

La información de cada línea deberá completar los pedidos de ambos arreglos (clientes y pedidos), buscando y seleccionando el cliente o producto correspondiente. **DEBE EMPLEAR OBLIGATORIAMENTE LOS OPERADORES SOBRECARGADOS.**

- d) Emitir un reporte con la información completa del almacén. El reporte debe tener un título y subtítulos adecuados.

De acuerdo con lo solicitado, la función "main" del proyecto estará compuesto por el siguiente código:

```
#include "Almacen.hpp"

using namespace std;

int main(int argc, char** argv) {
    Almacen almacen;

    almacen.cargar_clientes();
    almacen.cargar_productos();
    almacen.cargar_pedidos();
    almacen.mostrar_datos();

    return 0;
}
```

NO PUEDE CAMBIAR ESTE CÓDIGO

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, 27 de octubre del 2023.