

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

7ma 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**. **NO PUEDE UTILIZAR LA CLASE string**. Tampoco se podrán emplear las funciones de C que gestionen memoria como malloc, realloc, memset, strdup, strtok o similares, 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**
- **EL PROYECTO DEBERÁ SER DESARROLLADO BAJO EL PARADIGMA DE PROGRAMACIÓN ORIENTADO A OBJETOS POR LO QUE SALVO EN LA SOBRECARGA DE LOS OPERADORES >> Y <<, NO SE PODRÁN DEFINIR FUNCIONES (NI PLANTILLAS DE FUNCIONES) INDEPENDIENTES QUE NO ESTÉN LIGADAS COMO MÉTODOS A ALGUNA DE LAS CLASES PLANTEADAS.**
- Deberá mantener en todo momento el encapsulamiento de todos los atributos de las clases, esto implica también que un método NO puede devolver un puntero a algún dato del objeto. También debe guardar los estándares en la definición y uso de todas las clases desarrolladas. No deberá repetir código en los diferentes métodos. No se considerará en la nota las clases que violen esto.
- Cada método NO debe sobrepasar las 20 líneas de código aproximadamente.
- 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 un método si 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.
- **TAMPOCO SE PODRÁ EMPLEAR LA CLÁUSULA protected NI LA CLÁUSULA friend, DE HACERLO SE NO SE LE CALIFICARÁN LAS CLASES INVOLUCRADAS.**

SE LES RECUERDA QUE, DE ACUERDO CON EL 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_2024_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 7: "Herencia".

Descripción del Problema:

Se le ha solicitado desarrollar un sistema en C++ para gestionar y calcular **comandas** de atención en un restaurante. Las comandas son documentos que los camareros redactan para tomar nota de los diferentes pedidos de los clientes, dejando claro qué platos se van a pedir para cada cliente, y de esta forma comunicarlo correctamente a la cocina del restaurante. El objetivo es que cada comanda pueda registrar las bebidas, entradas y platos de fondo solicitados, calcular el total recaudado por la comanda y verificar si el tiempo de atención fue adecuado en función del tiempo de preparación estimado para cada tipo de producto. El tiempo de preparación estimado por producto se describe a continuación:

Bebida : 10 minutos, Entrada: 15 minutos, Plato de Fondo: 20 minutos

PARTE 01 (16 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto **"PREG01_LAB07"** dentro de la carpeta correspondiente, **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL**, en la cual se declaren las clases descritas con las relaciones necesarias, que permitan manipularlas empleando herencia:

- ❖ **Para manejar los productos:** La clase se denominará **"Producto"** y deberá contener lo siguiente: 1) un atributo denominado **nombre** (**char***) y 2) un atributo denominado **precio** (**double**).
- ❖ **Para manejar a los productos bebibles:** La clase se denominará **"Bebida"** y deberá contener lo siguiente: 1) un atributo denominado **tamano** (**char***), este campo almacena el tamaño de bebida que se puede solicitar (ej., "grande", "mediano"). Además, esta clase posee datos heredados de la clase **Producto**.
- ❖ **Para manejar a los productos de entrada:** La clase se denominará **"Entrada"** y deberá contener lo siguiente: 1) un atributo denominado **picante** (**bool**), donde se almacena si la entrada es picante o no. Además, esta clase posee datos heredados de la clase **Producto**.
- ❖ **Para manejar a los productos de platos de fondo:** La clase se denominará **"PlatoFondo"** y deberá contener lo siguiente: 1) un atributo denominado **proteina** (**int***), este atributo almacena un arreglo de 4 elementos en cuyas posiciones nos indican si la proteína (Índice 0: Pollo, Índice 1: Carne, Índice 2: Pescado, Índice 3: Lácteos) se encuentra incluida en el plato. Además, esta clase posee datos heredados de la clase **Producto**.
- ❖ **Para manejar una Comanda:** la clase se denominará **"Comanda"** y deberá contener lo siguiente: 1) un atributo **id** (**int**), 2) un atributo denominado **bebidas**, este atributo es un arreglo estático de la clase **Bebida**. 3) un atributo denominado **entradas**, este atributo es un arreglo estático de la clase **Entrada**. 4) un atributo denominado **platos_fondo**, este atributo es un arreglo estático de la clase **PlatoFondo**. 5) un atributo **cantidad_bebidas** (**int**), 6) un atributo **cantidad_entradas** (**int**) y 7) un atributo **cantidad_platos_fondo** (**int**), 8) un atributo **hora_atencion** (**int**), 9) un atributo denominado **hora_servicio** (**int**), 10) un atributo **tiempo_preparacion** (**int**), 11) un atributo **total** (**double**) y un 12) **estado** (**char***).
- ❖ **Para manejar todas las comandas:** La clase se denominará **"Restaurante"** y deberá contener lo siguiente: 1) un atributo denominado **comandas**, este atributo es un arreglo estático de la clase **Comanda**, donde se guardarán todas las comandas, 2) un atributo **cantidad_comandas** (**int**)

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

Con las clases indicadas debe realizar las siguientes operaciones:

- En la clase **Restaurante** debe implementar el método **cargar_comandas**, que se encarga de la lectura de los archivos "atenciones.csv" y "comandas.csv", con la finalidad de cargar en el arreglo de **comandas** la información brindada.

atenciones.csv		
1,	12:00,	12:20
2,	13:15,	13:40
3,	14:30,	14:50
...		

ID Comanda, hora de atención, hora de servicio.

comandas.csv					
1	B	Coca-Cola	3.5	grande	
1	E	Nachos	5.0	picante	
1	P	Lomo Saltado	12.0	0,1,0,0	
2	B	Sprite	3.0	mediano	
2	E	Guacamole	4.5	sin picante	
2	P	Pasta Alfredo	10.0	1,0,0,1	
3	B	Jugo de Naranja	4.0	pequeno	
3	E	Ensalada César	6.0	sin picante	
3	P	Filete de Pescado	11.0	1,0,0,0	
3	P	Stroganoff	25.75	0,1,0,1	
...					
ID Comanda, tipo de producto, nombre del producto, precio del producto, tamaño/picante/proteína.					

- En la clase **Restaurante** implementar el método **imprimir_comandas**, que se encargue de realizar la impresión de un archivo de prueba debidamente tabulado (**sin usar el carácter '\t'**), que muestre el contenido del arreglo **comandas** correspondientes a las comandas.

Consideraciones:

Para el desarrollo de ambas preguntas debe considerar el siguiente código, con excepción del método **actualizar_comandas** que solo debe estar en la pregunta 2:

```
#include "Restaurante.hpp"

using namespace std;

int main(int argc, char** argv) {
    Restaurante restaurante;

    restaurante.cargar_comandas();
    restaurante.actualizar_comandas();
    restaurante.imprimir_comandas();

    return 0;
}
```

**NO PUEDE
CAMBIAR
ESTE CÓDIGO**

Recuerde que si no usa herencia la respuesta no será válida.

PARTE 2 (4 puntos): Proceso final.

Desarrolle un proyecto denominado **"PREG02_LAB07"** en el cual se utilizarán obligatoriamente las clases desarrolladas en la pregunta anterior. El proyecto ejecutará las tareas descritas a continuación:

- Cargar el arreglo de **comandas** de acuerdo con lo indicado en la pregunta anterior.
- En la clase **Restaurante** debe implementar el método **actualizar_comandas**, que se encargará de calcular el valor **total** de la comanda y además modificar el **estado** de la comanda a "ATENDIDA" O "ATENDIDA CON RETRASO", según corresponda.
- Para que una comanda sea atendida con retraso, el tiempo de preparación debe ser mayor al tiempo transcurrido entre la hora de atención y la hora de servicio.

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.**

Profesores del curso:
Rony Cueva
Andrés Melgar
Miguel Guanira

Erasmus Gómez
Erick Huiza

San Miguel, 08 de Noviembre del 2024.