PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1
Examen 2
(Segundo Semestre 2019)

Indicaciones Generales:

- Duración: 3h.
- Se podrá usar como material de consulta solo sus apuntes de clase **(NO fotocopias, impresos ni hojas sueltas)**.
- No se pueden emplear **variables globales ni estructuras.** **Tampoco se podrá emplear la clase string, n**i 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. 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.
- La cláusula **friend** solo se podrá emplear en el caso de clases autoreferenciadas para ligar el nodo con la clase **inmediata** que encapsula la lista, **en ningún caso adicional**. No se considerará en la nota las clase que violen esto
- **Deberá guardar los estándares en la definición y uso de todas las clases desarrolladas.**
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada módulo **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 cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le **descontará 0.5 puntos por archivo**.
- Deberán respetar tanto los nombres de los proyectos como el de las clases y atributos. **Se descontará 0.5 pts. Por cada omisión.**
- **El código comentado NO SE CALIFICARÁ.**
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- **La presentación, la ortografía y la gramática de los trabajos influirá en la calificación.**

Puntaje total: 20 puntos

Cuestionario

PREGUNTA 1 (7 puntos)

Elabore un proyecto denominado "**Pregunta01PlantillaC**". El proyecto deberá definir una plantilla de clases (class PlCola) que permita definir una lista ligada que se comporte como una cola. La plantilla deberá permitir la llegada y la atención de un dato. Además permitirá mostrar, en un archivo de textos, los datos contenidos en la cola (desde el que esté al inicio de la cola, hasta el que esté al final, sin retirar los datos de la cola. La cola debe manejarse mediante dos punteros, uno al inicio y otro al final de la cola. Luego de atender todos los datos la cola debe quedar vacía, por lo que la atención de un dato eliminará el nodo que lo contenga. La impresión de los datos debe estar correctamente tabulada y con un título que explique la naturaleza de los datos (no podrá emplear el carácter '\t'). **La cola no podrá ser manipulada como una lista simplemente ligada**.

Para probar la plantilla deberá solucionar el problema que se describe a continuación. Deberá crear, a partir de la plantilla PlCola, una cola que permita trabajar con **objetos** de una clase denominada **Cliente**. Esta clase tendrá como atributos el DNI, nombre, hora de llegada y el tiempo que tomará la transacción que desea hacer. Los datos de cada cliente se leerán de un archivo de textos similar al que se muestra a continuación y se colocarán íntegramente en la cola. La prueba permitirá atender uno a uno los clientes y mostrar en un reporte, en un archivo de textos, los tiempos de espera de cada uno. El reporte será similar al que se muestra a continuación. El tiempo de espera dependerá de la hora en que se retire el cliente anterior, la hora de llegada del cliente y el tiempo de su transacción. La hora de salida del cliente depende de la hora en que se retire el cliente anterior y el tiempo de transacción (ver figura 1). La prueba se realizará a través de una clase denominada "**class Prueba**", que encapsulará todo el proceso de prueba.

Archivo de datos

```
10:00:00
02455871,Pérez/Rivas/Carlos Rodolfo,10:15:00,120.0
13456789,Castro/Quispe/María Fernanda,10:16:00,180.0
00445172,Fernandez/Zorrilla/Justino,10:55,23,90.00
…
```

La primera línea tiene la hora de apertura
El tiempo de transacción está en segundos

Reporte

```
Hora de apertura: 10:00:00

                                          HORA          TIEMPO      HORA
DNI       NOMBRE                          DE LLEGADA    DE ESPERA   DE SALIDA
=============================================================================
02455871  Pérez/Rivas/Carlos Rodolfo      10:15:00      120.00 seg. 10:17:00
13456789  Castro/Quispe/María Fernanda    10:16:00      240.00 seg. 10:20:00
00445172  Fernández/Zorrilla/Justino      10:55:23       90.00 seg. 10:56:53
…
=============================================================================
Promedio de espera: 185.56 seg.
```

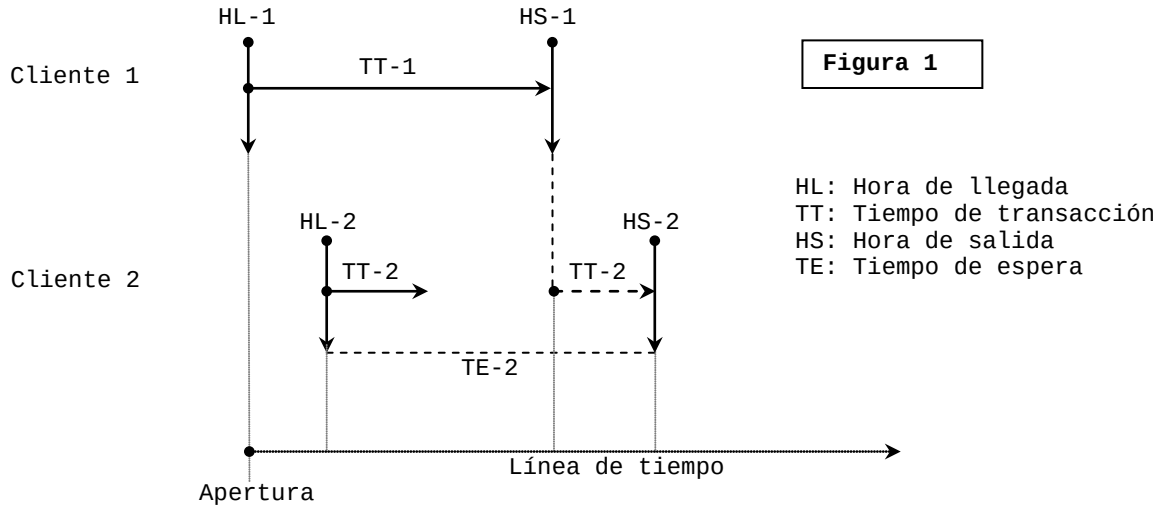


Figura 1

PREGUNTA 2 (3 puntos)

Se desea poder manipular tres tipos de pallet empleando polimorfismo. Las clase que los manejen tendrán un atributo que permite identificarlos (Id) que es un char*. Las clases permitirán determinar la capacidad del pallet, esto es, el peso que puede contener, el tipo de pallet y su Id o identificador.

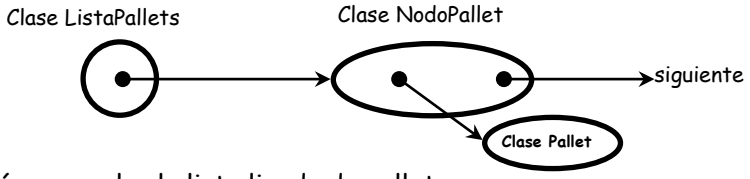
Los pallets según el tipo, siempre tienen la misma capacidad y se describen a continuación:

Pallet US: tiene una capacidad de 40 kg.
Pallet EU: tiene una capacidad de 50 kg.
Pallet PE: tiene una capacidad de 60 kg.

Deberá elaborar un proyecto denominado"**Pregunta02-Polimorfismo**" que definan las clases de modo que puedan ser manipuladas con polimorfismo. La implementación de estas clases contemplará la definición de atributos, constructores por defecto, destructores, métodos selectores, un método que permita leer los datos desde un archivo de texto y otro que permita imprimirlo en otro archivo de texto, y un método que devuelva el tipo de pallet ('U' para el pallet US, 'E' para el pallet EU y 'P' para el pallet PE). El proyecto deberá colocar diferentes pallets en un arreglo y luego manipular cada uno con polimorfismo.

PREGUNTA 3 (3 puntos)

Elabore un proyecto denominado **"Preg03ListaPallets"**, copie en él las clases desarrolladas en la segunda pregunta e incorpórelas al proyecto. Cualquier modificación que haga en este proyecto a las clases incorporadas no modificará la calificación de la preguntas anterior, **NO SE HARÁN EXCEPCIONES**. El proyecto debe contener las siguientes clases:



"Clase ListaPallets", permitirá encapsular la lista ligada de pallets.
"Clase NodoPallet", tendrá como atributos **un puntero** de la clase Pallet y el puntero siguiente.
"Clase Pallet", permitirá encapsular la información de un pallet. Como en el almacén existen tres tipos de pallets, los cuales pueden ser pallet US, pallet EU y pallet PE, se requiere que esta información se manipule obligatoriamente con polimorfismo, considerado la clase de la pregunta anterior.

El proyecto deberá ser capaz de armar una lista ligada de pallets a partir de los archivos que a continuación se describen, ordenada por el Id del pallet. También debe ser capaz de imprimir el contenido de toda la lista en un archivo de textos, según el formato que se indica más adelante, los datos deben aparecer correctamente tabulados y alineados (no podrá emplear el carácter '\t').

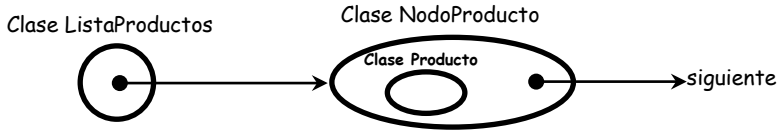
Los archivos de datos serán similares a los que se muestra a continuación:

pallets.csv

```
D41,EU
E10,US
B39,PE
A37,PE
.....
```

PREGUNTA 4 (4 puntos)

Elabore un proyecto denominado **"Preg04ListaProductos",** copie en él las clases desarrolladas en las preguntas anteriores e incorpórelas al proyecto. Cualquier modificación que haga en este proyecto a las clases incorporadas no modificará la calificación de la preguntas anterior, NO SE HARÁN EXCEPCIONES. El proyecto debe contener las siguientes clases:



"Clase ListaProductos", permitirá encapsular la lista ligada de productos.
"Clase NodoProducto", tendrá como atributos **un objeto** de la clase Producto y el puntero siguiente.
"Clase Producto", permitirá encapsular la información de un producto. La información del producto está conformado por:

- Un código único alfanumérico que representa el código del producto.
- Una lista de pallets que contienen el producto, debe usar la **clase pallet** desarrollada en las preguntas anteriores
- Una cola conformada por los pedidos del cliente a ser atendidos por producto, puede usar la plantilla desarrollada en la pregunta 1. La **clase Pedido** tiene como atributos un código del pedido numérico y la cantidad de productos solicitado en kg.
- Una cola de cantidades faltantes por pedido inicialmente vacía, puede emplearse la plantilla desarrollada en la pregunta 1. Esta cola estará conformada por la **clase Faltante** que tiene como atributos el código del pedido que presenta faltante y la cantidad de productos sin atender en kg.

El proyecto deberá ser capaz de armar una lista ligada de productos a partir del archivo **productos.csv**, ordenado por el código del producto. Luego se debe cargar la lista de stock de pallets empleando el archivo **stock.csv** También debe ser capaz de imprimir el contenido de cada una de las listas en archivos de textos, según el formato que se indica más adelante, los datos deben aparecer correctamente tabulados y alineados (no podrá emplear el carácter '\t'). Finalmente se debe cargar la cola de pedidos utilizando el archivo **pedidos.csv**, además de imprimir su contenido en un archivo de texto, pero sin desencolar los nodos (solo para este reporte se puede recorrer la cola como una lista con la finalidad de validar el contenido).

Los archivos de datos serán similares a los que se muestra a continuación:

stock.csv
CI87702,D41,EU
FZ98234,E10,US
HL94468,B39,PE
…..

productos.csv
CI87702
FZ98234
HL94468
…..

pedidos.csv
1002,SB26668,50
1003,FZ98234,60
1004,HL94468,110
…..

Pregunta 5 (2 puntos)

Elabore un método **Despacha** de la clase ListaProductos que realice la atención de los pedidos de toda la lista de Productos. Para cada producto, este método debe cargar la cola de faltantes a partir de las cantidades de la cola de pedidos, que no se lograron atender en su totalidad. Se sabe que los pedidos se realizan en cantidades que representan el peso de los pallets (40, 50 y 60). Al realizar la atención se debe desencolar el pedido y con la cantidad solicitada, se debe buscar en la **lista Pallets** de un determinado producto, la combinación que trate de atenderlo en su totalidad (el algoritmo para determinar la combinación queda a su criterio) **recuerde que no se pueden cortar o dividir los pallets**. Luego eliminar los nodos de la **lista de pallets** que se usaron para atender el pedido y las cantidades faltantes, deben ser encoladas en **faltantes**. Por ejemplo si consideramos el proceso para un producto determinado:

Pallets en stock: US, US, EU, PER, PER
Pedido: 180

Al despachar podría quedar de la siguiente forma:

Pallets en stock: US, US
Faltantes: 10

Para probar la operación imprimir en un archivo de texto, la cola de faltantes de cada producto, pero sin desencolar los nodos (solo para este reporte se puede recorrer la cola como una lista con la finalidad de validar el contenido). El reporte debe mostrar los faltantes de la siguiente forma:

Producto: SB26668
1002, 20
1003, 40

Pregunta 6 (1 punto)

Elabore un proyecto denominado **"Preg06SuperClase"**, copie las clases desarrolladas en las preguntas anteriores e incorpórelas al proyecto. Cualquier modificación que haga en este proyecto a las clases incorporadas no modificará la calificación de las preguntas anteriores, **NO SE HARÁN EXCEPCIONES**.

El proyecto debe contener una clase denominada "**clase Almacen**" la clase debe contener como único atributo un objeto (NO un puntero) de la clase listaProductos. La clase se encargará de abrir los archivos e invocar a los métodos que permitan cargar los productos, pallets de stock y los pedidos por cada producto y luego mostrar el reporte.

CONSIDERACIONES FINALES:

- Cree en el computador una carpeta de trabajo con la siguiente ruta: t:\temp\Examen2. En ella colocará los proyectos que den solución a los problemas planteados.
- De no respetarse el nombre de los proyectos y clases se descontará 1 punto por cada trasgresión.
- La calificación se otorgará por proyecto desarrollado. Por ninguna razón se asignará puntaje a dos o más preguntas por el mismo proyecto. **NO SE HARÁN EXCEPCIONES**
- Se prohíbe en esta evaluación el desarrollo de bibliotecas estáticas.

Al finalizar el examen, comprima[1] la carpeta ***Examen2*** en un archivo con nombre <código del alumno con 8 dígitos>.zip y súbalo a la intranet del curso, en el enlace Documentos, en la carpeta \Examen2\<código del horario>\<aula>. El acceso a la Intranet quedará cerrado automáticamente a las 11:05 am. por lo que el alumno que no suba alguno de los proyectos a la Intranet recibirá como nota CERO en esa pregunta. **NO SE HARÁN EXCEPCIONES.**

Profesores del curso: Rony Cueva
J. Miguel Guanira E. San Miguel, 10 de diciembre del 2019.

[1] Para evitar problemas en la corrección de la prueba, utilice el programa de compresión por defecto de Windows (Zip), **no use 7z**.