

Indicaciones específicas:

- Esta evaluación contiene 5 páginas (incluyendo esta página) con 1 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Las preguntas deberá ser respondida en un archivo fuente (.cpp) y un archivo cabecera (.h).

1. `p1.cpp`, `p1.h`

- Deberás subir estos archivos directamente a www.gradescope.com. También puedes crear un .zip

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación

Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)

Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)

Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)

- Para los alumnos de las carreras de Ingeniería

Capacidad de aplicar conocimientos de matemáticas (nivel 3)

Capacidad de aplicar conocimientos de ingeniería(nivel 2)

Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

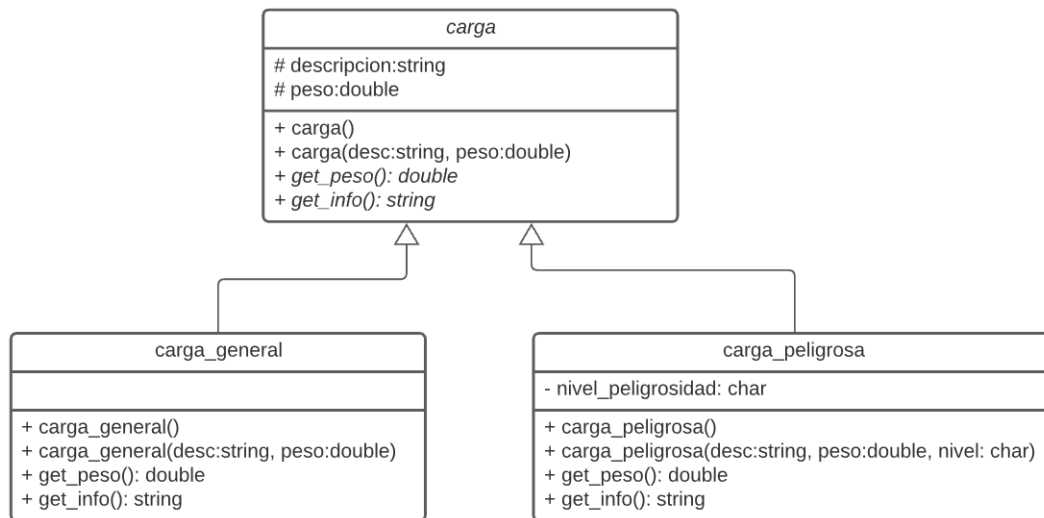
Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	20	
Total:	20	

1. (20 points) Escribir un programa que utilizando la clase **carga** y sus clases derivadas **carga_general** y **carga_peligrosa** genere un vector polimorfo, que permita almacenar **n** cargas, ingresando su tipo (**G=General** y **P=Peligrosa**), descripción, peso y en el caso de carga peligrosa el nivel de peligrosidad (**A=Alta**, **M=Mediana** y **B=Baja**), el programa debe de retornar la lista y el peso total de todas las cargas, para lo cual el resultado debe obtenerse utilizando el operador sobrecargado **<<** en el vector polimorfo (ver diagrama).

```
ostream& operator<<(ostream& out, vector<carga*> cargas);
```



Ejemplo 1: Input

```

7
G Carga1 10
G Carga2 5
P Carga3 2 A
G Carga4 5
P Carga5 2 M
G Carga6 5
P Carga7 2 B
  
```

Output

```

Carga1 10
Carga2 5
Carga3 2 A
Carga4 5
Carga5 2 M
Carga6 5
Carga7 2 B
Total = 31
  
```

La rúbrica para esta pregunta sobre **Herencia** es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Código	Ha implementado herencia de clases correctamente implementando adecuadamente los atributos y sus niveles de acceso, los constructores han sido adecuadamente creados en las clases derivadas (3pts)	Existen algunos errores menores en la implementación (2pts)	El diseño y la implementación del código no son correctos (0pts).
Sintaxis	No existen errores sintácticos o de compilación de la solución (3pts)	Existen algunos errores sintácticos o de compilación. (1pts).	El código no compila (0pts).
Optimizacion	El código es óptimo y eficiente de la solución (1pts)	El código es optimizable en algunas partes (0.5pts).	El código es redundante y/o no es óptimo (0pts).

La rúbrica para esta pregunta sobre **Polimorfismo** es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Código	Ha implementado clases virtuales y sobreescritura adecuadamente, se ha implementado un repositorio polimorfo que permite almacenar una clase base y puede invocar adecuadamente los métodos sobreescritos de la clase derivada (3pts)	Existen algunos errores menores en la implementación (2pts)	El diseño y la implementación del código no son correctos (0pts).
Sintaxis	No existen errores sintácticos o de compilación de la solución (3pts)	Existen algunos errores sintácticos o de compilación. (1pts).	El código no compila (0pts).
Optimizacion	El código es óptimo y eficiente de la solución (1pts)	El código es optimizable en algunas partes (0.5pts).	El código es redundante y/o no es óptimo (0pts).

La rúbrica para esta pregunta sobre **Sobrecarga de Operadores** es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Código	Ha definido clases y se ha implementado la sobrecarga de operadores y funciones como se ha solicitado (3pts)	Existen algunos errores menores en la implementación (2pts)	El diseño y la implementación del código no son correctos (0pts).
Sintaxis	No existen errores sintácticos o de compilación de la solución (2pts)	Existen algunos errores sintácticos o de compilación. (1pts).	El código no compila (0pts).
Optimizacion	El código es óptimo y eficiente de la solución (1pts)	El código es optimizable en algunas partes (0.5pts).	El código es redundante y/o no es óptimo (0pts).