

Indicaciones específicas:

- Esta evaluación contiene 7 páginas (incluyendo esta página) con 1 único caso que tiene 2 sub-preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Todo el código deberá estar en el archivo main.cpp y este será el único archivo que se debe entregar como solución a www.gradescope.com
- Respete los nombres de las clases, atributos y métodos sugeridos.
- **Recuerda que solo se calificará si cumples con estas indicaciones.**

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
 - Aplicar conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa. (nivel 2)
 - Diseñar, implementar y evaluar soluciones a problemas complejos de computación. (nivel 2)
 - Crear, seleccionar, adaptar y aplicar técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones. (nivel 2)
- Para los alumnos de las carreras de Ingeniería
 - Aplicar conocimientos de ingeniería en la solución de problemas complejos de ingeniería. (nivel 2)
 - Diseñar soluciones relacionados a problemas complejos de ingeniería. (nivel 2)
 - Crear, seleccionar y utilizar técnicas, habilidades, recursos y herramientas modernas de la ingeniería y las tecnologías de la información, incluyendo la predicción y el modelamiento, con la comprensión de sus limitaciones. (nivel 2)
- Para los alumnos de Administración y Negocios Digitales
 - Analizar información verbal y/o lógica proveniente de distintas fuentes, encontrando relaciones y presentándola de manera clara y concisa (nivel 2)
 - Analizar y evaluar el comportamiento del consumidor y el desarrollo de estrategias comerciales (nivel 2)
 - Trabajar de manera efectiva con equipos multidisciplinarios y diversos en género, nacionalidad, edad, etc. (nivel 2)

Calificación:

Tabla de puntos (solo para uso del profesor)

Question	Points	Score
<u>1</u>	20	
Total:	20	

1. (20 points) Procesador de exámenes de admisión

Realizar un programa que permita gestionar el proceso de examen de admisión de un grupo de postulantes a cierta universidad.

Por cada postulante se tiene registrado la siguiente información:

- Facultad a la que postula (L – Letras o C – Ciencias)
- DNI
- Apellidos y Nombres
- Número de preguntas de lectura respondidas correctamente
- Número de preguntas de redacción respondidas correctamente
- Número de preguntas de matemática respondidas correctamente

La calificación del postulante depende de la facultad a la que se postula:

Facultad	Calificación (puntos)
Letras	<ul style="list-style-type: none">• 7.5 por cada pregunta de lectura y redacción respondida correctamente• 10 por cada pregunta de matemática respondida correctamente
Ciencias	<ul style="list-style-type: none">• 10 por cada pregunta de lectura y redacción respondida correctamente• 7.5 por cada pregunta de matemática respondida correctamente

La información de todos los postulantes se encuentra en un archivo de texto (postulantes.csv) con el siguiente formato:

Facultad,DNI,Apellidos y Nombres,Lectura,Redacción,Matematica

Cada línea representa los datos de un postulante separado por comas (.). Por ejemplo:

C,75323355,Zarate Valverde José,20,20,80,

Indica:

C → Indica que es postulante es de Ciencias

75323355 → DNI del postulante

Zarate Valverde José → Apellidos y Nombres del postulante

20 → Preguntas de lectura respondidas correctamente

20 → Preguntas de redacción respondidas correctamente

80 → Preguntas de matemática respondidas correctamente

Para que un postulante pueda ingresar a la universidad debe alcanzar como mínimo 750 puntos de calificación.

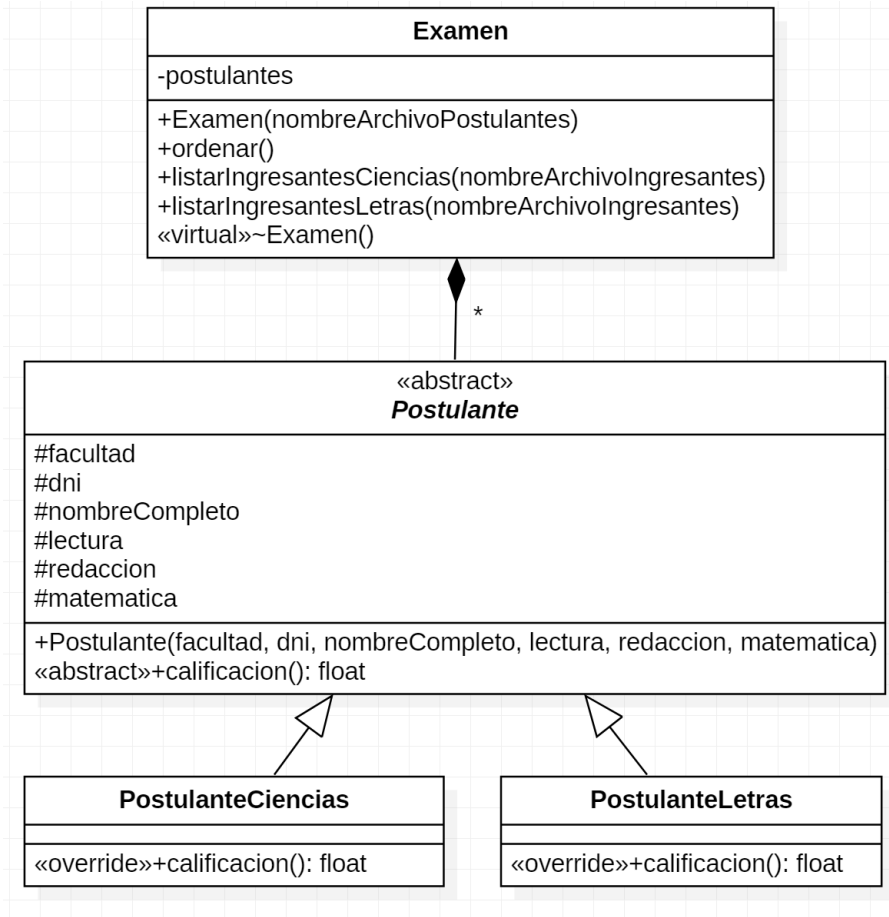
El programa deberá procesar toda la información contenida en el archivo de texto y generar los resultados en dos archivos diferentes, uno para ciencias (ciencias.csv) y otra para letras (letras.csv) con el siguiente formato: Facultad,DNI,Apellidos y Nombres,Calificacion.

Para el ejemplo mencionado párrafos arriba al resultado sería:

C,75323355,Zarate Valverde Jose,1000.000000,

Los resultados deberán aparecer ordenados por la calificación obtenida en forma descendente. En caso de que haya postulantes con el mismo puntaje deberá ordenarse por apellidos y nombres.

La solución debe considerar el uso de Clases Abstractas, Herencia, Polimorfismo y Archivos. Use como referencia el siguiente diagrama de clases:



Se solicita desarrollar el **Programa Orientado a Objetos** considerando lo siguiente:

1. Implementación de **Herencia y Polimorfismo** en jerarquía de clases **Postulante**, **PostulanteCiencias** y **PostulanteLetras**. Si considera necesario añada atributos y métodos a las clases sugeridas **(7.5p)**.
2. Implementación de la clase **Examen**. Si considera necesario añada atributos y métodos de modo que cubra la siguiente funcionalidad **(12.5p)**
 - Lectura de archivo y creación de objetos **PostulanteCiencias** o **PostulanteLetras** a partir de cada línea del archivo (3.0 p).
 - Ordenamiento Postulantes por calificación obtenida y nombres (3.0 p)
 - Listado de Ingresantes de Ciencias en el archivo indicado (3.0 p)
 - Listado de Ingresantes de Letras en el archivo indicado (3.0 p)
 - Liberación de objetos en el destructor (0.5 p).

Algunos ejemplos de interacción del programa serían:

Archivo de entrada:

Listing 1: postulantes.csv

```
Facultad,DNI,Apellidos y Nombre,Lectura,Redaccion,Matematica,  
L,23272090,Contreras Chavez Estanislao,40,40,5,  
C,75323355,Zarate Valverde Jose,20,20,80,  
L,34567890,Saavedra Zapata Milena,25,23,10,  
C,12344567,Merino Capa Zulema,10,12,60,  
C,20191035,Aguado Iriarte Martin Samuel,18,18,70,  
L,20182007,Aguirre Bernal Alvaro Gonzalo,20,12,14,  
L,20171051,Bartle Tesoro Christian Alexander,12,16,40,  
C,20182003,Berrocal Alvarado Willian ,14,13,4,  
C,20181035,Cabezas Baquerizo Cesar Enrique,15,20,7,  
L,20171026,De loayza Espejo Ivan Rodrigo,10,12,25,  
L,20192015,Enciso Lozano Diego Sebastian,19,14,23,  
C,20182009,Heredia Lapa Cristhoper Michel,20,25,75,  
C,20171035,Imparato Conetta Massimo Nicola,5,13,12,  
L,20191051,Jauregui Vera Luis Aaron Eduardo,3,15,20,  
C,20171007,Juscamaita Lopez Jean Paul,5,11,40,  
L,20161059,Lapa Carhuamaca Arleth Ivhy,12,3,50,  
C,20181004,Merma Silva Carlos Antonio,7,2,12,  
L,20191040,Nakamura Montenegro Akira Sebastian,6,6,14,  
C,20181005,Osterling Dankers Kurt Rafael,20,13,74,  
C,20171048,Oyague Santolalla Macarena Milagros,3,8,78,  
L,20181022,Quispe Barriga Sebastian Andres,4,9,23,  
L,20191041,Quispe Estrada Romulo Gherson,19,14,12,  
C,20171063,Rojas Rojas Christian Kevin,10,12,50,  
C,20182014,Salas Palacios Eduardo Miguel,15,12,12,  
C,20141029,Tolentino Mendivil Angel Alberto,13,11,13,  
C,20181053,Vilela Mori Tomas,12,1,15,  
C,20181070,Villanueva Perea Diego Jose,5,3,30,
```

Archivo de salida para ciencias:

Listing 2: ciencias.csv

```
Facultad,DNI,Apellidos y Nombre,Calificacion
C,20182009,Heredia Lapa Cristhoper Michel,1012.500000
C,75323355,Zarate Valverde Jose,1000.000000
C,20191035,Aguado Iriarte Martin Samuel,885.000000
C,20181005,Osterling Dankers Kurt Rafael,885.000000
C,20171048,Oyague Santolalla Macarena Milagros,695.000000
C,12344567,Merino Capa Zulema,670.000000
C,20171063,Rojas Rojas Christian Kevin,595.000000
C,20171007,Juscamaita Lopez Jean Paul,460.000000
C,20181035,Cabezas Baquerizo Cesar Enrique,402.500000
C,20182014,Salas Palacios Eduardo Miguel,360.000000
C,20141029,Tolentino Mendivil Angel Alberto,337.500000
C,20181070,Villanueva Perea Diego Jose,305.000000
C,20182003,Berrocal Alvarado Willian ,300.000000
C,20171035,Imparato Conetta Massimo Nicola,270.000000
C,20181053,Vilela Mori Tomas,242.500000
C,20181004,Merma Silva Carlos Antonio,180.000000
```

Archivo de salida para letras:

Listing 3: letras.csv

```
Facultad,DNI,Apellidos y Nombre,Calificacion
L,75323355,Contreras Chavez Estanislao,650.000000
L,20161059,Lapa Carhuamaca Arleth Ivhy,612.500000
L,20171051,Bartle Tesoro Christian Alexander,610.000000
L,20192015,Enciso Lozano Diego Sebastian,477.500000
L,34567890,Saavedra Zapata Milena,460.000000
L,20171026,De loayza Espejo Ivan Rodrigo,415.000000
L,20182007,Aguirre Bernal Alvaro Gonzalo,380.000000
L,20191041,Quispe Estrada Romulo Gherson,367.500000
L,20191051,Jauregui Vera Luis Aaron Eduardo,335.000000
L,20181022,Quispe Barriga Sebastian Andres,327.500000
L,20191040,Nakamura Montenegro Akira Sebastian,230.000000
```

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado y no tenga errores de compilación.

El 100% corresponde al puntaje indicado en cada punto

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución (40%)	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (100%)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (70%)	El diseño tiene algunas deficiencias pero la ejecución es correcta (30%).	El diseño es deficiente y la ejecución no es correcta (0%)
Sintaxis (30%)	No existen errores sintácticos o de compilación (100%)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (50%).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (30%).	El código tiene errores de sintaxis que afectan el resultado (10%).
Optimización (30%)	El código es óptimo y eficiente. De buen performance e interacción con el usuario (100%)	El código es de buen performance durante la ejecución (70%).	El código no está optimizado pero la ejecución no es deficiente(30%).	El código no está optimizado y la ejecución es deficiente (0%).