

#### Indicaciones específicas:

- Esta evaluación contiene 4 páginas (incluyendo esta página) con 1 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- El código debe consistir de un **main.cpp** y archivos de encabezado **.h**. Si lo requiere, también archivos **.cpp** adicionales
- Deberá subir estos archivos directamente a [www.gradescope.com](http://www.gradescope.com), por separado o en un **.zip**
- Se solicita activar cámara durante la evaluación. En caso de contingencia, justifique por correo electrónico . . . @utec.edu.pe

#### 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 (sólo para uso del professor)

Question	Points	Score
1	20	
Total:	20	

1. (20 points) **Centro de enseñanzas**

Desarrolle los siguientes puntos en un modelo **Orientado a objetos** para simular la interacción de un Centro de enseñanza:

1. Genere una clase **CentroEnseñanza**, con atributo(s) **Nombre**, **Direccion** y **Vector de Alumnos**. La clase **Alumno** tiene los atributos(s) **Edad**, **Nombre**, **Carretera** y **vector de Cursos**. La clase **curso** está conformado por **NombreCurso**, **Creditos**, **PromedioFinal**. (4 pts)
2. Asimismo la clase **CentroEnseñanza** deriva **UParticular**, con atributo **Mensualidad** y una clase derivada **UEstatal** con atributo **AñoInauguración**. La clase **CentroEnseñanza** tiene un método **ImprimirDatosBasicos** y las clases derivadas deben implementarlo de forma distinta. (4 pts)
3. Ahora genera 1 objeto de tipo **UParticular** con 5 Alumnos con 3 Cursos como mínimo. Para el caso de **UNacional**, crear 1 objeto con 7 alumnos y 4 cursos como mínimo. Para agregar los cursos a los alumnos se debe hacer uso de sobre carga del operador **<<**. Mostrar el promedio de notas por Centro de enseñanza. (5 pts)
4. Defina una método que determine el promedio de notas de los alumnos por universidad, el cual se implemente utilizando sobrecarga del operador **+** para sumar los promedios de las notas de los alumnos. (3 pts)
5. Crear un documento .txt en el cual. se guarde de la siguiente forma.

Nombre Universidad,cantidad alumnos, promedio notas de alumnos.  
UTEC,5,12.5  
UNMSM,7,10.5

Solicitar al usuario el nombre del archivo .txt (4 pts)

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100% corresponde al puntaje indicado en cada punto

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	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	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%)
Optimizacion	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%)