UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Programación II
Práctica Calificada 3
Pregrado
2022-I
Profesor: José Fiestas
Lab: 1.03

Indicaciones específicas:

- Esta evaluación contiene 3 páginas (incluyendo esta página) con 1 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- El código puede consistir de un solo archivo **main.cpp**; o bien un **.cpp** y archivos de encabezado **.h**. Si lo requiere, tambien archivos **.cpp** adicionales
- Deberá subir estos archivos directamente a www.gradescope.com, por separado o en un **.zip**
- Se solicita activar cámara durante la evaluación. En caso de contingencia, justifique por correo electrónico a jfiestas@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)

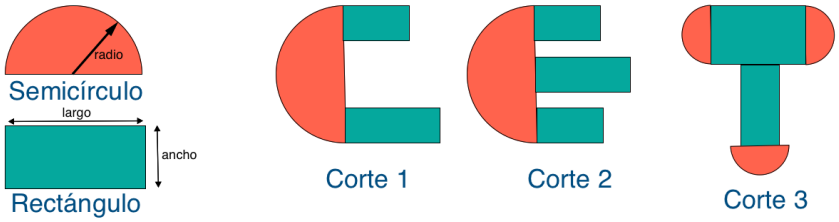
Question	Points	Score
1	20	
Total:	20	

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 sintáxis que afectan el resultado (10 %)
Complejidad	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 %)

1. [20 puntos]

   Para la elaboración de juguetes de madera, se requiere de piezas de corte transversal que están hechas de piezas básicas (rectángulos y semicírculos)



   1. Implemente una clase **Pieza**, con, por lo menos, un atributo para la **cantidad de piezas a producirse**, así como un método para calcular su **área (2 pts)**
   2. Una clase derivada **Rectangulo**, que contenga atributos **largo** y **ancho**, y un método para calcular su **área (2 pts)**
   3. Una clase derivada **Semicírculo**, que contenga un atributo **radio** y un método para calcular su **área (2 pts)**
   4. Defina una clase para el **Corte Transversal** que almacene las piezas básicas en un **vector** y permita calcular su área. La cantidad de piezas debe ser como en las figuras pero pueden agregarse en cualquier orden **(3 pts)**
   5. Sobrecargue el operador + para agregar piezas básicas a un **Corte Transversal**.- Es decir, que se pueda agregar una pieza como: *corte+pieza;* **(2 pts)**
   6. Programe la elaboración de: un juguete hecho de 3 cortes como en la figura 1, 2 cortes como en la figura 2, y 1 corte como en la figura 3 **(3 pts)**
   7. Sobrecargue el operador $\ll$ para imprimir los datos del juguete: cantidad, tipo y área de cada corte, y área total del juguete **(3 pts)**
   8. Implemente un método para modificar el diseño de un juguete, reemplazando una pieza básica de un **Corte Transversal**. Por ejemplo, cambiando un semicírculo por un rectángulo. Muestre su funcionalidad con un ejemplo, e imprima de nuevo los datos del juguete **(3 pts)**