Programación Orientada a Objetos 1

Práctica Calificada 3

Pregrado

2021-I

Profesor: Rubén Rivas Medina

Lab 1.01

Indicaciones específicas:

- Esta evaluación contiene 7 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. `pc3.cpp`, `pc3.h`

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

Competencias y criterios de desempeño:

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

    Aplica conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa.(nivel 2)

    Diseña, implementa y evalúa soluciones a problemas complejos de computación.(nivel 2)

    Crea, selecciona, adapta y aplica 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

    Capacidad para aplicar conocimientos de matemática.(nivel 2)

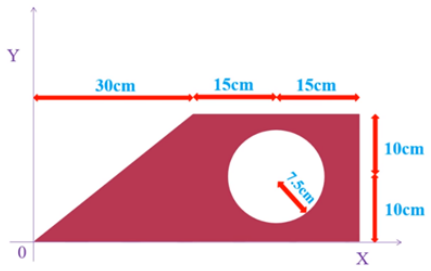
    Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas(nivel 2)

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	20	
Total:	20	

1. (20 points) Implementa un programa orientado a objetos que permite calcular el area plana compuesta como la que se muestra a continuacion:

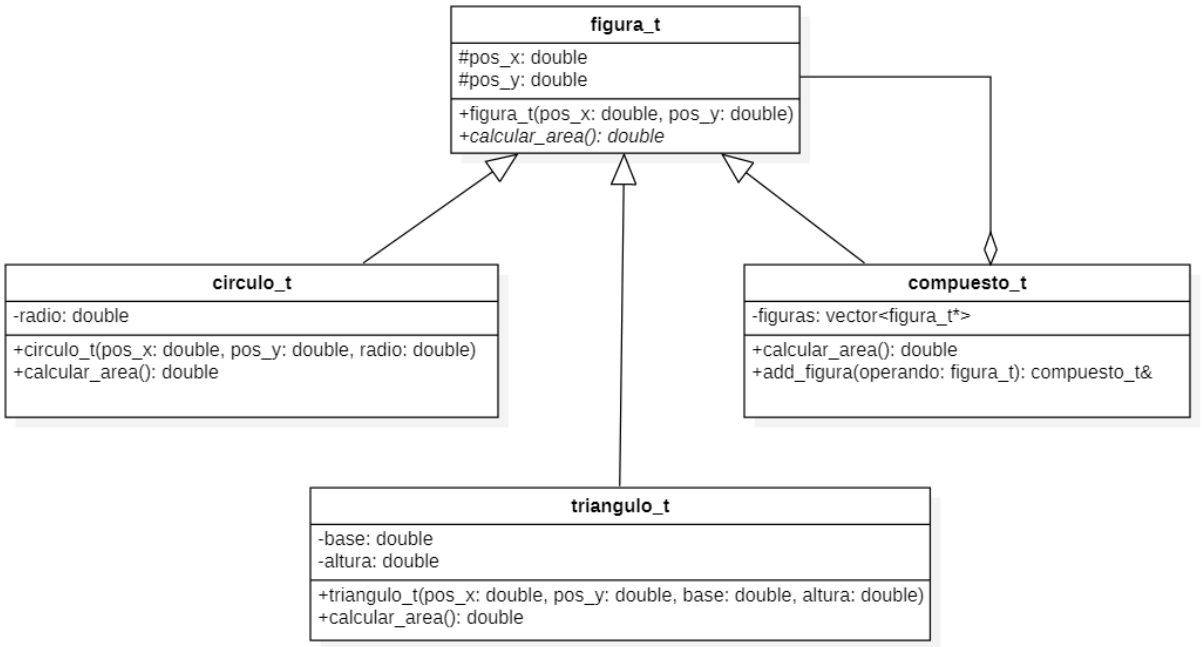


- La impresión del area de la figura compuesta debe realizarse por medio de la sobrecarga del operador desplazamiento a la izquierda **(ver ejemplo)**.
- Agregar a la clase **compuesto_t** sobrecarga al operador incremento (+=) basandose en el metodo **add_figura** de la misma clase **(ver diagrama)**

```
ostream& operator<<(ostream& out, compuesto_t comp);
```

Nota: El programa debe permitir calcular el área de solo figuras compuestas como se muestra en la imagen.

Sugerencias: Basarse en el diagrama de clases mostrado a continuación y agregar al programa la clase rectangulo:



Ejemplo 1: Input

```
T 0 0 30 20
R 30 0 30 20
C 37.5 2.5 7.5
```

Output

```
1076.71
```

Ejemplo 2: Input

```
R 50 0 50 40
T 0 0 50 40
C 60 5 15
```

Output

```
3706.86
```

La rúbrica para esta pregunta sobre **relaciones y herencia** es:

Criterio	**Excelente**	**Adecuado**	**Mínimo**	**Insuficiente**
Relaciones entre clases y/o usa polimorfismo **(5 pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Establece relaciones de: composición, agregación, herencia**, y al implementar las clases lo hace con el 100% de precisión. **(5pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Establece relaciones de: composición, agregación, herencia**, y al implementar las clases lo hace con al menos el 80% de precisión. **(3pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Establece relaciones de: composición, agregación, herencia**, y al implementar las clases lo hace con al menos el 65% de precisión. **(2pts)**	El programa es orientado a objetos, pero no da solución exacta a lo que el enunciado requiere. **Establece relaciones de: composición, agregación, herencia**, y al implementar las clases lo hace con menos el 65% de precisión. **(0pts)**
Bloque Principal del programa **(3 pt)**	Crea objetos instanciando las clases y los usa de acuerdo a lo solicitado.**(3pts)**	Crea objetos instanciando las clases pero no los usa adecuadamente **(2pts)**	Trata de crear los objetos pero tiene errores que afectan el resultado del programa. **(1pts)**	No crea objetos, ni lo usa de manera adecuada. **(0pts)**

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

Criterio	**Excelente**	**Adecuado**	**Mínimo**	**Insuficiente**
Relaciones entre clases, archivos, sobrecarga de operadores **(4 pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Establece relaciones entre clases polimorfismo**. Al implementar las clases lo hace con el 100% de precisión. **(4pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Establece relaciones entre clases polimorfismo**. Al implementar las clases lo hace con al menos el 80% de precisión. **(2pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Establece relaciones entre clases polimorfismo**. Al implementar las clases lo hace con al menos el 65% de precisión. **(1pts)**	El programa es orientado a objetos, pero no da solución exacta a lo que el enunciado requiere. **Establece relaciones entre clases polimorfismo**, y al implementar las clases lo hace con menos el 65% de precisión. **(0pts)**
Bloque Principal del programa **(2 pt)**	Crea objetos instanciando las clases y los usa de acuerdo a lo solicitado.**(2pts)**	Crea objetos instanciando las clases pero no los usa adecuadamente **(1pts)**	Trata de crear los objetos pero tiene errores que afectan el resultado del programa. **0.5pts)**	No crea objetos, ni lo usa de manera adecuada. **(0pts)**

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

Criterio	**Excelente**	**Adecuado**	**Mínimo**	**Insuficiente**
Relaciones entre clases, archivos, sobrecarga de operadores **(4 pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Utiliza sobrecarga de operadores**. Al implementar las clases lo hace con el 100% de precisión. **(4pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Utiliza sobrecarga de operadores**. Al implementar las clases lo hace con al menos el 80% de precisión. **(2pts)**	El programa es orientado a objetos y da solución exacta a lo que el enunciado requiere. **Utiliza sobrecarga de operadores**. Al implementar las clases lo hace con al menos el 65% de precisión. **(1pts)**	El programa es orientado a objetos, pero no da solución exacta a lo que el enunciado requiere. **Utiliza sobrecarga de operadores**, y al implementar las clases lo hace con menos el 65% de precisión. **(0pts)**
Bloque Principal del programa **(2 pt)**	Crea objetos instanciando las clases y los usa de acuerdo a lo solicitado.**(2pts)**	Crea objetos instanciando las clases pero no los usa adecuadamente **(1pts)**	Trata de crear los objetos pero tiene errores que afectan el resultado del programa. **0.5pts)**	No crea objetos, ni lo usa de manera adecuada. **(0pts)**