

Segundo Parcial 2021

Parte teórica

- 1) ¿Qué utilidad tiene un constructor de una *clase abstracta*, si no se pueden crear objetos de la misma?
- 2) ¿Cuáles son los tipos de *enlace* que se pueden establecer en C++ entre la declaración y la definición de un método? y ¿Qué relación existe con el *polimorfismo*?
- 3) ¿Explique brevemente con sus palabras que es un *Template* (Planilla) de C++ y cuando es apropiado utilizarlo?
- 4) ¿Qué es una *excepción* y como se maneja en C++? Ejemplifique.
- 5) ¿Cuáles son las diferencias entre una *clase abstracta* y una *interface*?

Parte práctica

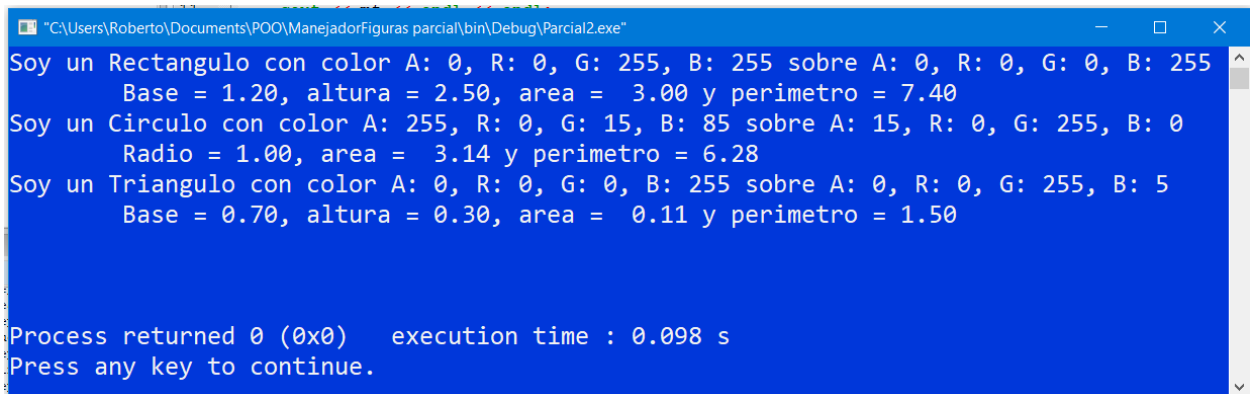
1) Escriba un programa en lenguaje C++ que permita ejecutar el conjunto de sentencias o tareas mostradas en el código que se encuentra en el archivo *main.cpp*, que se adjunta. Se aclara que el código que se entrega es mínimo, debiendo agregar al mismo las instrucciones que crea necesarias para que el programa se ejecute sin que termine en forma anormal.

Básicamente, en un programa que permite la gestión de figuras de distinto tipo y colores (tanto de base como de fondo) y en función de sus dimensiones (base y altura) permite obtener el área y el perímetro de cada una. En el caso de un círculo, considerar que la primer dimensión corresponde al radio y la otra siempre es 0.

Se suministran además dos archivos adicionales con la declaración y la definición de una clase auxiliar (*CColor*) que permite el manejo de los colores, tanto en su formato entero como en sus 4 componentes: alfa, rojo, verde y azul.

El programa comienza recuperando los datos almacenados en un archivo de texto que contiene la información de cada figura en una nueva línea y dentro de la misma se utiliza como separador un punto y coma (‘;’) de la forma: *tipo_figura;fore_color;back_color;base;altura*. En ningún momento se conoce la cantidad de figuras con que se trabaja (no se necesita tampoco).

Una vez levantados los datos se deben presentar en una consola con el siguiente formato:



```
"C:\Users\Roberto\Documents\POO\ManejadorFiguras parcial\bin\Debug\Parcial2.exe"
Soy un Rectangulo con color A: 0, R: 0, G: 255, B: 255 sobre A: 0, R: 0, G: 0, B: 255
    Base = 1.20, altura = 2.50, area = 3.00 y perimetro = 7.40
Soy un Circulo con color A: 255, R: 0, G: 15, B: 85 sobre A: 15, R: 0, G: 255, B: 0
    Radio = 1.00, area = 3.14 y perimetro = 6.28
Soy un Triangulo con color A: 0, R: 0, G: 0, B: 255 sobre A: 0, R: 0, G: 255, B: 5
    Base = 0.70, altura = 0.30, area = 0.11 y perimetro = 1.50

Process returned 0 (0x0)   execution time : 0.098 s
Press any key to continue.
```

Cada vez que se agregue una nueva figura se debe guardar la nueva información en un archivo de texto con el mismo formato que se utilizó en el archivo de entrada, de forma que en futuras invocaciones del programa pueda ser utilizado para recuperar la información de TODAS las figuras.

Finalmente se deben ordenar los datos de acuerdo al **criterio** que se envía como parámetro a la función específica *ordenar*(nombre_archivo, **criterio**). Si criterio = “area”, las figuras se deben ordenar de acuerdo a su área en forma creciente. Si criterio = “perimetro” entonces deben ordenarse de acuerdo a su perímetro en forma decreciente. La información en el archivo de salida se debe presentar: *tipo_figura, area = 123.45 y perimetro = 456.78<CR>*

Se debe realizar una correcta modularización del código, y el programa debe construirse (build) sin errores (errors) ni advertencias (warnings). Como se mencionó el programa no debe terminar en forma anormal ante un error en tiempo de ejecución.