

Demonstration of SOLID principles

Develop a Visual Studio CONSOLE APPLICATION which simulates a drawing package. Your application should support the following 5 drawing primitives (we'll call them widgets):

- 1) Rectangle
- 2) Square
- 3) Ellipse
- 4) Circle
- 5) Textbox

Requirements

The application should allow widgets to be added to the drawing, stating the location and size/shape of the widget.

The location can be a standard x/y coordinate on an imaginary page. The size/shape depends on the widget, as follows:

- Rectangle – width and height
- Square – width
- Ellipse – horizontal and vertical diameter
- Circle – diameter
- Textbox – bounding rectangle (i.e., the rectangle which surrounds the textbox; the text will be centred within this rectangle).

Clarifications

- Integer units are fine for all dimensions.
- Widths, heights and diameters must be positive.
- For the textbox you only need to configure the text to display (which will be rendered horizontally within the bounding rectangle). Don't worry about font face /size / alignment, etc.
- You DO NOT need to concern yourself with handling any input from the user. Our widgets will be hard-coded within the Main() method.
- Your console application should be able to 'print out' the current drawing by printing the key details of each widget (type, location, size/shape) to the console.
- You don't need to actually render the widgets in any manner - we're just simulating a drawing package at this stage.
- **You must add one of each widget to the drawing as per the next page**

Important:

Unit Testing: Please include appropriate unit tests with your solution.

Readme: Finally, explain your design decisions in a **readme** file alongside your submitted solution. Also mention any further work you would do if you had more time.

Submission: To submit the exercise, please ensure that all the project files and folders are uploaded in a zip file. Ideally, exclude the bin and obj directories as the presence of executables could risk a delay in your submission being processed due to antivirus processing.

Grading: Your exercise will be evaluated on your ability to follow requirements while demonstrating good programming practices such as: Maintainability, Extensibility, Robustness to changing requirements, Testability and SOLID principles. If you're invited to an interview you will be asked to explain your design choices and you will be asked to amend your solution in response to a requirement change.

Simulation:

In your Main method, ensure that the following widgets are added.

```
rectangle x=10, y=10, width=30 height=40
square x=15, y=30, size=35
ellipse x=100, y=150, horizontal diameter=300, vertical diameter=200
circle x=1, y=1, size=300
textbox x=5, y=5, width=200, height=100, text="sample text"
```

Required Output

As a result, when running the application, it must produce the following output

Requested Drawing

```
Rectangle (10,10) width=30 height=40
Square (15,30) size=35
Ellipse (100,150) diameterH = 300 diameterV = 200
Circle (1,1) size=300
Textbox (5,5) width=200 height=100 Text="sample text"
-----
```