Industrial Software Development (ISDe)

Exercise 1

USE THE DOUBLE DISPATCH APPROACH

Use abstract classes if needed.

Consider a program in which different geometric **shapes** can be drawn on different types of **displays**.

For simplicity we associate only the **name** to each geometric shape (we omit the properties of the geometric shape, its coordinates, etc. that we will have to indicate in a real problem)

```
    r = Rectangle('**blue rectangle**')
    c = Circle('**big red circle**')
```

For simplicity, we associate only the **name** with each display (we omit the hardware properties, resolution, etc. that we will have to indicate in a real problem)

In the example we have a type 1 display (**d1**) and two type 2 displays (**d2_a, d2_b**)

```
    d1 = Display1('800x600x8')
    d2_a = Display2('1600x1600x16 - A')
    d2_b = Display2('1600x1600x16 - B')
```

Requirements:

The specific algorithm for drawing the geometric shape resides in the class of the geometric shape itself. For simplicity, the **draw()** method merely prints a string:

```
draw the rectangle  **blue rectangle**  into the display of type 1 : 800x600x8
```

The method to draw the shape is a method of the **display** object.

Example

```
    d1.draw(c)
    d2_a.draw(r)
```

Write a module **my_graph.py** that can be used with the assigned **main.py**. Complete the **main.py** if needed.
The assigned **main.py** contains (as a comment) the correct, expected output.

Be sure to submit both the **main** and the **module**. Your program must work correctly and produce the correct output.