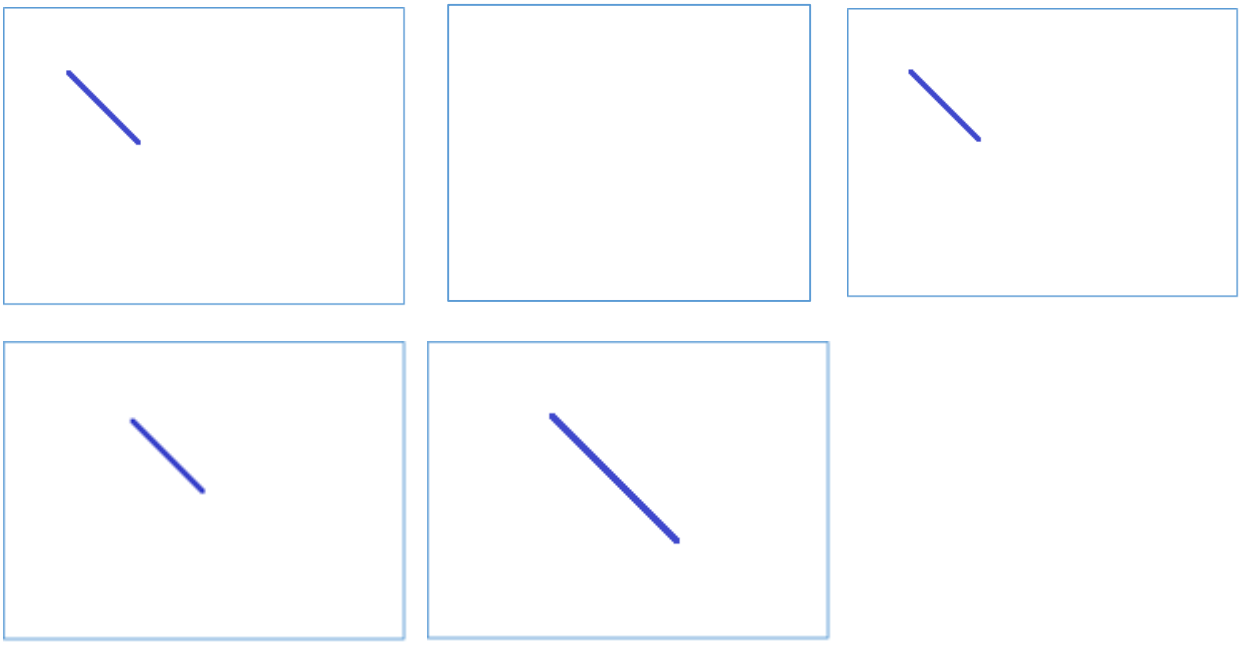
Practice 18 [Inheritance-2]

1. Consider the drawing code share in last class. The shared code *shape*, *line* & *circle* classes. Modify shape class, add another data member ***background color*** (default value 'WHITE'). Add following member functions for overriding in shape class:
   - ***remove*** – no parameter, simply redraw same shape in background color
   - ***move*** – with two parameter x, y having new reference position of the shape. Redraw shape at existing position in background color. Set data members and redraw shape in foreground color at new position
   - ***change_size*** – with one parameter that is change of size in percentage. For example, in case of +20, for a circle shape, increase radius by 20%. For example, if radius was 50, now it will be 60. Again, redraw shape in background color before change. Change data member to adjust size and redraw shape in foreground color according to new values

   Modify *Line* class accordingly. Override three functions added in Shape class and run following driver program:

```
import pygame as py
from circle import *
from line import * from time import *
def main():
    screen = py.display.set_mode((800, 600))
    screen.fill('white')
    py.display.update()
    line = Line(screen, 100, 100, 200, 200)
    line.draw()
    sleep(1 )
    line.remove()
    sleep(1)
    line.draw()
    sleep(1)
`    line.move(220, 130)
    sleep(1)
    line.increase_size(50)
```



There are five frames captured from output of the code. In upper left frame draw function of line is called. In upper middle function remove function of line is called. In upper right frame, draw function of line is called again. In lower left frame, move function of line is called. Finally, in lower right frame increase_size function of line is called.

Next, modify Circle class and run following driver code:
import pygame as py

```
from circle import *
from line import *
from time import *

def main():
    screen = py.display.set_mode((800, 600))
    screen.fill('white')
    py.display.update()
    #line = Line(screen, 100, 100, 200, 200)
    circle = Circle(screen, 450, 250, 50)
    circle.draw()
    sleep(1)
    circle.remove()
    sleep(1)
    circle.draw()
    sleep(1)
    circle.move(250, 150)
    sleep(1)
    circle.increase_size(30)

main()
```

2. Create another child class ***Hollow_Circle*** derived (inherit) from ***Circle*** class. Add getter, setter function in *circle* class to get and set values of data members. Copy paste functions from *Circle* class and do following changes:

   First of all, add another parameter in draw circle function that is a constant value 1 to draw hollow circle. Next, to access data members of *Circle* class, call getter setters, like this:

   ```
   py.draw.circle(self.screen, self.color, super().get_center(), super().get_radius(),1)
   super().set_center(x, y)      #In move function
   super().set_radius( super().get_radius() * ( 100 + size ) // 100 ) #In increase_size function
   ```

3. Create Shaded Circle class derived from Circle class. Try to use multiple colors to draw circle. The idea is start drawing color with variation in radius. Start from larger radius towards smaller radius.

4. Create a polygon class derived from shape. The polygon class has list of points. Draw function with draw lines between them such that first point will be connected with second, second with third and so on. Finally, last point will be connected to first point to close the shape.