



Homework #4

**01286121 Computer Programming
Software Engineering Program,
Department of Computer Engineering,
School of Engineering, KMITL**

By

66011149 Phatthadon Sornplang

- 1.) Write a program that prompts the user to enter the x- and y-coordinates for the three points p0, p1, and p2, and the program then displays a message to indicate whether p2 is on the left side of, the right side of, or on the line between p0 and p1.

```
import turtle as t
```

```
x0 = int(input("Enter Point #0 x coordinate:"))
```

```
y0 = int(input("Enter Point #0 y coordinate:"))
```

```
x1 = int(input("Enter Point #1 x coordinate:"))
```

```
y1 = int(input("Enter Point #1 y coordinate:"))
```

```
x2 = int(input("Enter Point #2 x coordinate:"))
```

```
y2 = int(input("Enter Point #2 y coordinate:"))
```

```
determine = (x1 - x0) * (y2 - y0) - (x2 - x0) * (y1 - y0)
```

```
write_ans = (x2 + 50, y2 - 50)
```

```
t.penup()
```

```
t.goto(x0, y0)
```

```
t.pendown()
```

```
t.write("p0")
```

```
t.goto(x1, y1)
```

```
t.write("p1")
```

```
t.penup()
```

```
t.goto(x2, y2)
```

```
t.pendown()
```

```
t.dot(5)
```

```
t.write("p2")
```

```
t.penup()
```

```
t.goto(write_ans)
```

```
t.pendown
```

```
if determine > 0 :
```

```
    t.write("p2 is on the left side of the line")
```

```
elif determine < 0:
```

```
    t.write("p2 is on the right side of the line")
```

```
elif determine == 0:
```

```
t.write("p2 is on the line")
```

```
t.hideturtle()
```

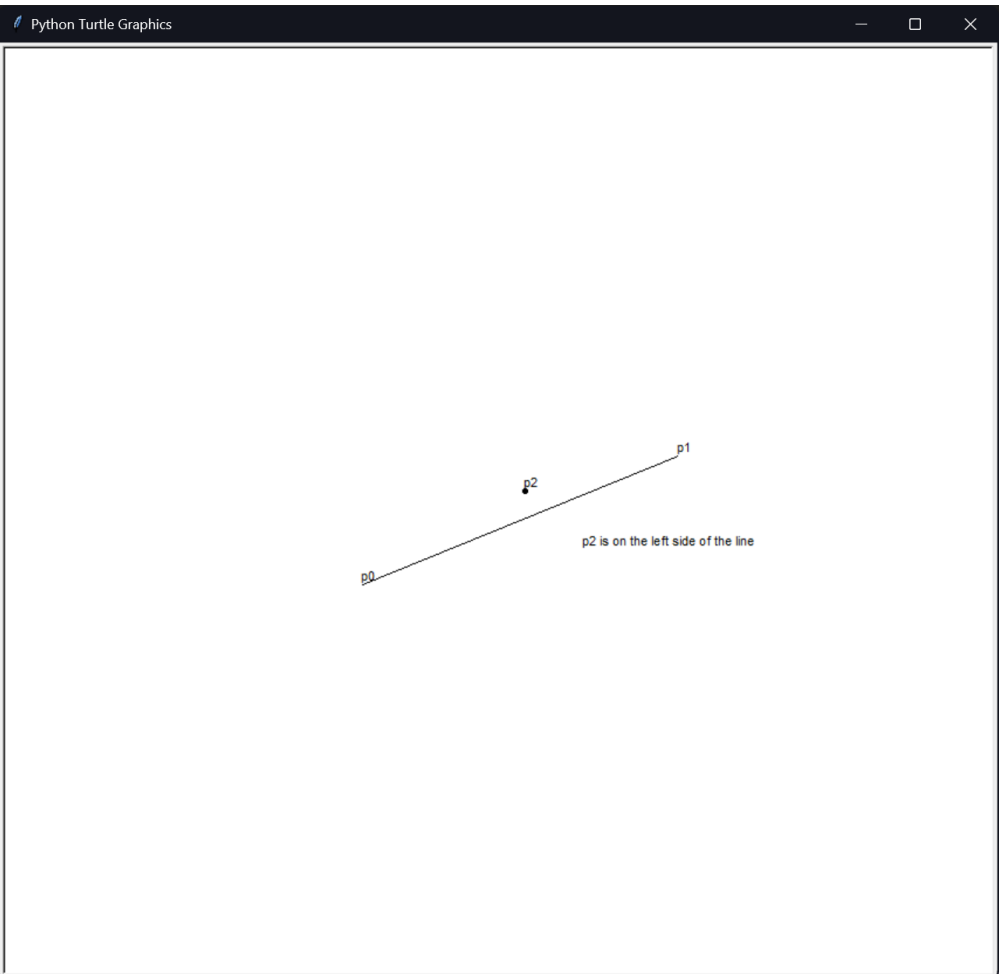
```
t.done()
```

```
20 t.goto(x2, y2)
21 t.pendown()
22 t.dot(5)
23 t.write("p2")
24 t.penup()
25 t.goto(write_ans)
26 t.pendown
27 if determine > 0 :
28     t.write("p2 is on the left
29 elif determine < 0:
30     t.write("p2 is on the righ
31 elif determine == 0:
32     t.write("p2 is on the line
33
34 t.hideturtle()
35 t.done()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Enter Point #1 x coordinate:200
Enter Point #1 y coordinate:50
Enter Point #2 x coordinate:50
Enter Point #2 y coordinate:60
```

```
phatt@Macbook_Pro MINGW64 ~/OneDrive/De
$ C:/Users/phatt/AppData/Local/Programs
on)/4/HW/HW4_1.py"
Enter Point #0 x coordinate:-120
Enter Point #0 y coordinate:-60
Enter Point #1 x coordinate:150
Enter Point #1 y coordinate:50
Enter Point #2 x coordinate:19
Enter Point #2 y coordinate:20
█
```



- 2.) Write a program that prompts the user to enter the center x-, y-coordinates, width and height of two rectangles and the program then determines whether which rectangle is inside or overlap with the other.

```
import turtle as t
```

```
x1 = int(input("Enter Rectangle #1 x coordinate:"))
```

```
y1 = int(input("Enter Rectangle #1 y coordinate:"))
```

```
w1 = int(input("Enter Rectangle #1 Width:"))
```

```
h1 = int(input("Enter Rectangle #1 Height:"))
```

```
x2 = int(input("Enter Rectangle #2 x coordinate:"))
```

```
y2 = int(input("Enter Rectangle #2 y coordinate:"))
```

```
w2 = int(input("Enter Rectangle #2 Width:"))
```

```
h2 = int(input("Enter Rectangle #2 Height:"))
```

```
if x1 - x2 >= 0:
```

```
    x_distance = x1 - x2
```

```
else:
```

```
    x_distance = x2 - x1
```

```
if y1 - y2 >= 0:
```

```
    y_distance = y1 - y2
```

```
else:
```

```
    y_distance = y2 - y1
```

```
if x_distance <= (w1 - w2)/2 and y_distance <= (h1 - h2)/2:
```

```
    print("Rectangle #2 is inside Rectangle #1")
```

```
elif x_distance <= (w1 + w2)/2 and y_distance <= (h1 + h2)/2:
```

```
    print("Rectangle #2 overlaps Rectangle #1")
```

```
else:
```

```
    print("Rectangle #2 does not overlap Rectangle #1")
```

```
t.penup()
```

```
t.goto(x1, y1)
```

```

t.write("Rec #1" + "(" + str(x1) + "," + str(y1) + ")")

t.goto(x1 - (w1 / 2), y1 + (h1 / 2))

t.pendown()

t.goto(x1 + (w1 / 2), y1 + (h1 / 2))

t.goto(x1 + (w1 / 2), y1 - (h1 / 2))

t.goto(x1 - (w1 / 2), y1 - (h1 / 2))

t.goto(x1 - (w1 / 2), y1 + (h1 / 2))


t.penup()


t.goto(x2, y2)

t.write("Rec #2" + "(" + str(x2) + "," + str(y2) + ")")

t.goto(x2 - (w2 / 2), y2 + (h2 / 2))

t.pendown()

t.goto(x2 + (w2 / 2), y2 + (h2 / 2))

t.goto(x2 + (w2 / 2), y2 - (h2 / 2))

t.goto(x2 - (w2 / 2), y2 - (h2 / 2))

t.goto(x2 - (w2 / 2), y2 + (h2 / 2))

t.penup()

t.goto(x2 + w2, y2 + h2)

if x_distance <= (w1 - w2)/2 and y_distance <= (h1 - h2)/2:

    t.write("Rectangle #2 is inside Rectangle #1")

elif x_distance <= (w1 + w2)/2 and y_distance <= (h1 + h2)/2:

    t.write("Rectangle #2 overlaps Rectangle #1")

else:

    t.write("Rectangle #2 does not overlap Rectangle #1")


t.hideturtle()

t.done()

```

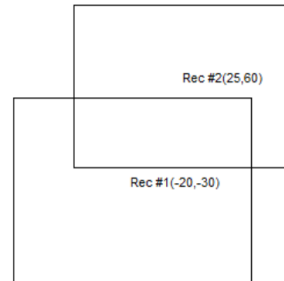
```

31 t.goto(x1, y1)
32 t.write("Rec #1" + "(" + str(x1) + ", " + str(y1) + ")")
33 t.goto(x1 - (w1 / 2), y1 + (h1 / 2))
34 t.pendown()
35 t.goto(x1 + (w1 / 2), y1 + (h1 / 2))
36 t.goto(x1 + (w1 / 2), y1 - (h1 / 2))
37 t.goto(x1 - (w1 / 2), y1 - (h1 / 2))
38 t.goto(x1 - (w1 / 2), y1 + (h1 / 2))
39
40 t.penup()
41
42 t.goto(x2, y2)
43 t.write("Rec #2" + "(" + str(x2) + ", " + str(y2) + ")")
44 t.goto(x2 - (w2 / 2), y2 + (h2 / 2))
45 t.pendown()
46 t.goto(x2 + (w2 / 2), y2 + (h2 / 2))
47 t.goto(x2 + (w2 / 2), y2 - (h2 / 2))
48 t.goto(x2 - (w2 / 2), y2 - (h2 / 2))
49 t.goto(x2 - (w2 / 2), y2 + (h2 / 2))
50 t.penup()
51 t.goto(x2 + w2, y2 + h2)
52 if x_distance <= (w1 - w2) / 2

```

Python Turtle Graphics

Rectangle #2 overlaps Rectangle #1



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

phatt@Macbook_Pro MINGW64 ~/OneDrive/...
$ C:/Users/phatt/AppData/Local/Program...
on)/4/HW/HW4_2.py"
Enter Rectangle #1 x coordinate:-20
Enter Rectangle #1 y coordinate:-30
Enter Rectangle #1 Width:205
Enter Rectangle #1 Height:160
Enter Rectangle #2 x coordinate:25
Enter Rectangle #2 y coordinate:60
Enter Rectangle #2 Width:190
Enter Rectangle #2 Height:140
Rectangle #2 overlaps Rectangle #1

```