In-Class Question 10

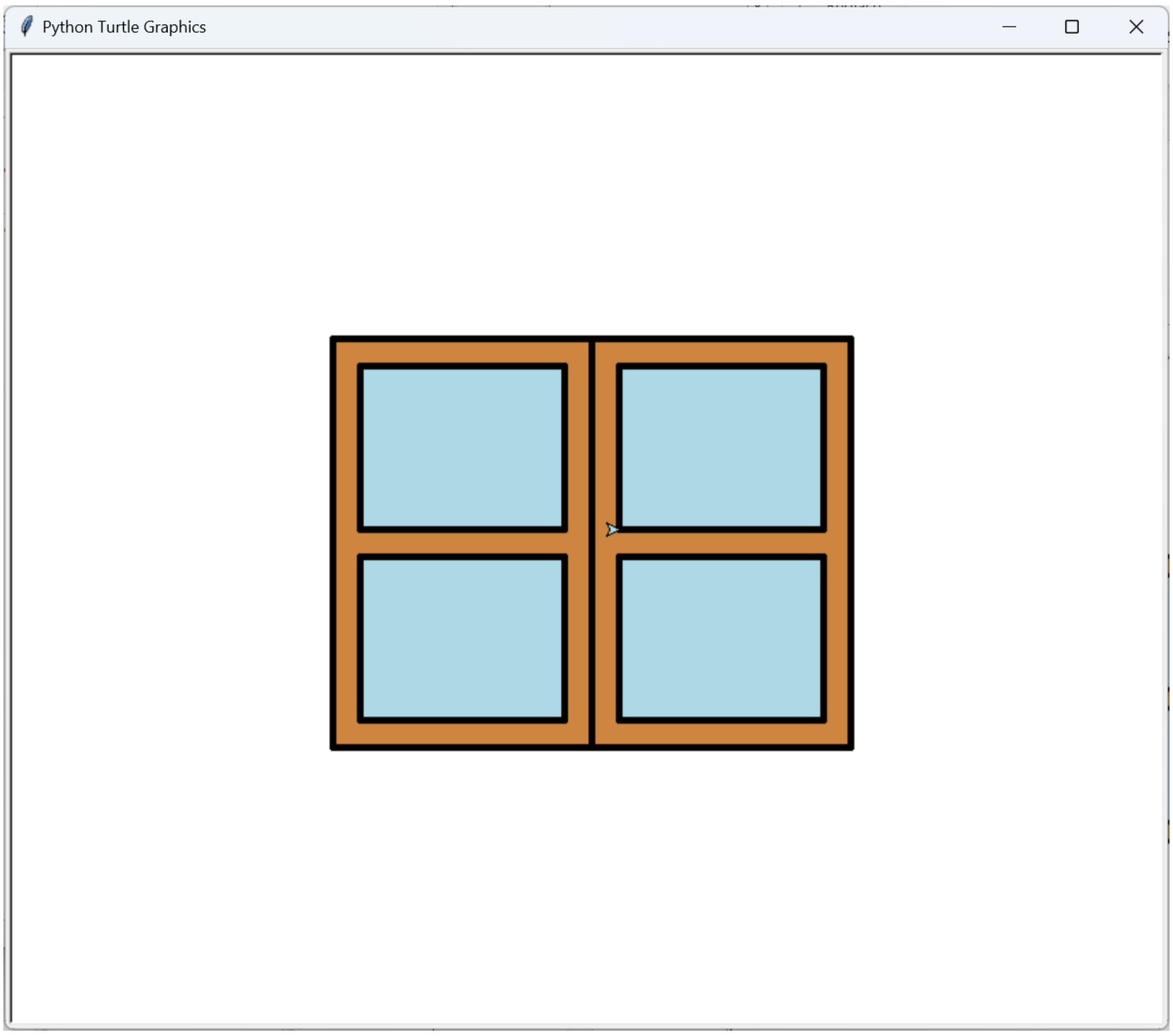
Question)

The Python program shown below draws a cartoon window using turtle graphics programming.

```
import turtle

turtle.speed(0)
turtle.width(5)

def draw_window(x, y, color, w, h):
    turtle.up()
    turtle.goto(x, y)
    turtle.down()

    turtle.color("black", color)

    turtle.begin_fill()
    for _ in range(2):
        turtle.forward(w)
        turtle.left(90)
        turtle.forward(h)
        turtle.left(90)
    turtle.end_fill()

draw_window(190, -150, "peru", -190, 300)
draw_window(0, -150, "peru", -190, 300)

# The four pieces of glass are drawn in this order and
# positions:
# 1) x = -170, y = -130
# 2) x = -170, y = 10
# 3) x = 20, y = -130
# 4) x = 20, y = 10
for pos in SEE_QUESTION :
    draw_window(pos[0], pos[1], "lightblue", 150, 120)

turtle.done()
```

As you can see, the content of **`SEE_QUESTION`** is not shown in the above code.

The output of the program is shown below.



What is the missing code that needs to replace **`SEE_QUESTION`**? You need to enter the missing code. You must complete the code following the comments written in the program.

Correct answer(s):

- `[[-170, -130], [-170, 10], [20, -130], [20, 10]]`
- `([-170, -130], [-170, 10], [20, -130], [20, 10])`
- `[(-170, -130), (-170, 10), (20, -130), (20, 10)]`
- `((-170, -130), (-170, 10), (20, -130), (20, 10))`

Explanation:

- Because the for loop draws four pieces of glass, i.e. four light blue rectangles, you need to make sure the loop can run four times
- One way to do that is to make a list containing four items
- By looking at the way the `pos` variable is used inside the loop (`pos[0]` and `pos[1]`), you can see that the variable is a sequence data type, which can be a list, a tuple or a string
- Because the `pos` variable contains the x and y position, you can either use a list or a tuple
- Therefore, the missing code can be a list containing four items, with each item being a list as well, i.e.:

  ```
  [ [ ... ], [ ... ], [ ... ], [ ... ] ]
  ```

- If you want to, you can replace each of the above lists by a tuple as lists and tuples behave the same in the above situation
- The content of each item inside the list is then the x and y position of the rectangles, e.g. `[-170, -130]`