BIS 420 PROGRAMMING FOR DATA SCIENCE

PRAJAKTA POHARE
CHAPTER 15 EXERCISE 15.3
ILLINOIS STATE UNIVERSITY

Write a version of move_rectangle that creates and returns a new Rectangle instead of modifying the old one.

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

class Rectangle:
    def __init__(self, width, height, corner):
        self.width = width
        self.height = height
        self.corner = corner

def move_rectangle(rect, dx, dy):
    new_corner = Point(rect.corner.x + dx, rect.corner.y + dy)
    return Rectangle(rect.width, rect.height, new_corner)

original_corner = Point(0, 0)
original_rectangle = Rectangle(100, 50, original_corner)

moved_rectangle = move_rectangle(original_rectangle, 10, 20)

print(f"Original corner: ({original_rectangle.corner.x}, {original_rectangle.corner.y})")
```

print(f"Moved corner: ({moved_rectangle.corner.x}, {moved_rectangle.corner.y})")

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

class Rectangle:
    def __init__(self, width, height, corner):
        self.width = width
        self.height = height
        self.corner = corner

def move_rectangle(rect, dx, dy):
    new_corner = Point(rect.corner.x + dx, rect.corner.y + dy)
    return Rectangle(rect.width, rect.height, new_corner)


original_corner = Point(0, 0)
original_rectangle = Rectangle(100, 50, original_corner)

moved_rectangle = move_rectangle(original_rectangle, 10, 20)

print(f"Original corner: ({original_rectangle.corner.x},
{original_rectangle.corner.y})")
print(f"Moved corner: ({moved_rectangle.corner.x}, {moved_rectangle.corner.y})")
```