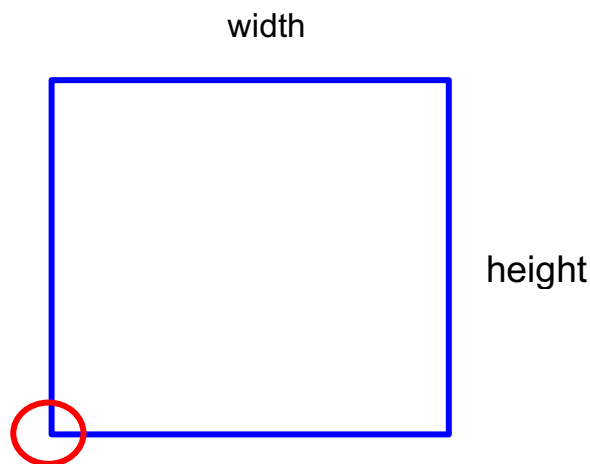


Week 5 Lab exercises

Exercise 1.

- Add a method called `insert_point_at(self, x, y, index)` to `Polygon` that inserts a new point after the given index. If the index is -1, then it should insert the point at the beginning of the list of points. If the index is greater than or equal to the length of the list, it should insert the point at the end.
- Write a `Rectangle` class which is a subclass of `Polygon`. However, the `Rectangle` is initialized by the left lower corner which is an instance of the `Point` class, the width and height, which is different to that of `Polygon`.



The left lower corner: p

That is the `__init__()` of `Rectangle` should look like the following:

```
def __init__(self, width, height, p):
```

You need to find a way to initialize the `Rectangle` properly so that the methods it inherits from the `Polygon` can work without any modification.

Exercise 2 – Car Class

Write a class named `Car` that has the following data attributes:

- `__year_model`(for the car's year model)
- `__brand`(for the car's brand)
- `__speed`(for the car's current speed **per hour**)

The `Car` class should have a `__init__` method that accepts the car's year model and make as arguments. It should also assign 0 to the `__speed` data attribute.

The class should also have the following methods:

- Accelerate: The `accelerate` method add 5 to the speed data attribute.
- Brake: The `brake` method subtract 5 from the speed data attribute.
- The `get_speed` method should return the current speed

Next, Design a program that creates a car object, and then calls the `accelerate` method 5 times, then `brake` 5 times. After each call display the current speed.

The travel time is broken into 10 *random* length segments (10 random integers of hours), at the end of each segment, there is acceleration or break; first 5 accelerations and then 5 breaks. Output the total distance traveled.

Bonus: Now make the action at the end of a segment a random choice.