MODULE 1 LAB ACTIVITIES
PART 2: IMPLEMENTING CLASSES

Objectives:

- Implement classes in Python.
- Gain practical understanding of interfaces and inheritance.
- Understand how to submit assignments to CodePost.

Repl.it Mini-Project

1. Open the Module 1 Lab Activities Part 2 assignment on Repl.it. The workspace contains the following files:

    - `MobileCritter.py` - contains the `MobileCritter` interface.
    - `Insect.py` - contains a fully-implemented superclass called `Insect`.
    - `Ant.py` - contains an incomplete subclass of `Insect` that must implement the `MobileCritter` interface.
    - `Spider.py` - contains an incomplete subclass of `Insect` that must implement the `MobileCritter` interface.
    - `Board.py` - contains a fully-implemented class called `Board`.
    - `main.py` - contains the complete implementation of a game that uses `Ant` and `Spider` objects.

2. Implement the `Ant` and `Spider` classes so that they fulfill the operations described in the documentation of the listed methods.
3. Test your implementation by running `main.py`. If your classes contain bugs, correct them.
4. Once you are satisfied with your implementation, download the `Ant.py` and `Spider.py` files to your computer.
5. Read the directions for submitting on CodePost that are found under Canvas » Modules» Housekeping » Project Resources » Submitting Your Work to CodePost. Then, submit the following files to the appropriate CodePost folder:

    - `Ant.py`
    - `Spider.py`

CodePost will run unit tests on your work and grade your submission according to the following rubric:

	Full Credit 1 Pt	Partial Credit Pts vary. See CodePost.	No Credit 0 Pt
`Ant` implementation	Implementation is correct and passes all CodePost tests.	Implementation is partially correct. Fails one or more CodePost tests	Implementation is incorrect/incomplete and fails all CodePost tests.
`Spider` implementation	Implementation is correct and passes all CodePost tests.	Implementation is partially correct; fails one or more CodePost tests.	Implementation is incorrect/incomplete and fails all CodePost tests.

Python Syntax

Submit a written response for the following questions:

1. (1 pt) Research: What is the purpose of `from ...` in each import statement of the header to the `main.py` file?

```
from Spider import Spider
from Ant import Ant
from Board import Board
```

2. (1 pt) Research: Notice that both the `Ant` and `Spider` classes seem to take arguments `Insect` and `MobileCritter` in the declaration of the class:

```
class Ant(Insect, MobileCritter):
    ...

class Spider(Insect, MobileCritter):
    ...
```

What is this syntax actually indicating?

Score: / 4 pts