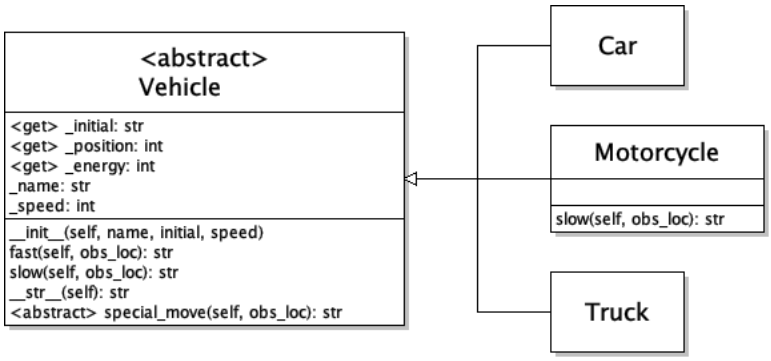
CECS 277 – Lab 8 – Abstract Classes

Rad Racer

Create a game where the user must choose a vehicle, either a Car, Motorcycle, or Truck and then race against the remaining choices. On each turn, a vehicle may choose to go fast, slow, or to use a special move. The vehicles race on a track that has obstacles in the way. If the vehicle is going fast, then it will crash, if it is going slow, then it can maneuver around the obstacle. The special moves are "Nitro Boost" for the car, which makes the car go 1.5x faster, "Wheelie" for the motorcycle, which makes it go 2x faster, but has a chance of falling over, and "Ram" for the truck, which makes it go 2x faster and also allows it to bash through an obstacle. Whichever vehicle reaches the finish first wins the race.
Use inheritance to implement the following class diagram in your program:



Vehicle Class – Abstract class - (in a separate file named: 'vehicle.py') –
Attributes: _name – vehicle name, _initial – vehicle's label, _speed – vehicle speed, _position – vehicle's current location on the track, _energy – vehicle's power level. Use decorators to create the properties for the initial, the position, and energy attributes.
Methods:

1. `__init__(self, name, initial, speed)` – set the attributes using the parameters, assign 0 to position and 100 to energy.
2. `fast(self, obs_loc)` – passes in the location of the next obstacle (if there is one). If there is sufficient energy (>= 5), randomize a value between +/- 1 speed for the number of spaces they will move and deduct 5 energy. If that movement is less than the distance to the next obstacle, then they move that amount, otherwise, it crashes into the obstacle and stops on the obstacle's space. If there wasn't sufficient energy, move one space forward. In any case, return a string that describes the event that occurred with the name of the vehicle and the distance traveled (if applicable).
3. `slow(self, obs_loc)` – passes in the location of the next obstacle (if there is one). The vehicle will move at half speed +/- 1. If there is an obstacle, then it will go around it. There is no energy cost. Return a string that describes the event that occurred with the name of the vehicle and the distance traveled.
4. `__str__(self)` – return a string with the vehicle's name, position, and energy.
5. `special_move(self, obs_loc)` – abstract method - overridden in the subclasses. Use a decorator to make them abstract.

Car Class – inherits from Vehicle – (in a separate file named: 'car.py') –

1. `special_move(self, obs_loc)` – nitro boost – passes in the location of the next obstacle (if there is one). If there is sufficient energy (>= 15), deduct 15 energy and move the car at 1.5x speed +/- 1. If there is an obstacle, then it will crash and stops on that space, otherwise it moves the randomized distance. If there was not sufficient energy, then only move up one space. Return a string that describes the event that occurred with the name of the car and the distance traveled (if applicable).

Motorcycle Class – inherits from Vehicle – (in a separate file named: 'motorcycle.py') –

1. `slow(self, obs_loc)` – overridden method - passes in the location of the next obstacle (if there is one). The motorcycle will move at 75% speed +/- 1, rather than half. If there's an obstacle, then it will go around it. There is no energy cost. Return a string that describes the event that occurred with the name of the motorcycle and the distance traveled (if applicable).
2. `special_move(self, obs_loc)` – passes in the location of the next obstacle (if there is one). If there is sufficient energy (>= 15), deduct 15 energy, then there is a 75% chance that the motorcycle will move at 2x speed +/- 1, otherwise it will fall over and only move one space forward. If it was successful but there is an obstacle, then it will crash and stops in that space, otherwise it moves the randomized distance. If there wasn't sufficient energy, then only move one space. Return a string that describes the event that occurred with the name of the motorcycle and the distance traveled (if applicable).

Truck Class – inherits from Vehicle – (in a separate file named: 'truck.py') –

1. `special_move(self, obs_loc)` – passes in the location of the next obstacle (if there is one). If there is sufficient energy (>= 15), deduct 15 energy and move the truck 2x speed, even if there is an obstacle ('ram' bashes through the obstacle). If there wasn't sufficient energy, then only move one space. Return a string that describes the event that occurred with the name of the truck and the distance traveled (if applicable).

Main (in a separate file named: 'main.py') – Construct a 2D list that stores the track. It should have three lanes, one for each vehicle and should be 100 units long. Randomly place two obstacles in each lane, but don't place them at the start or finish. Display the descriptions for each of the vehicles and prompt the user to choose one to play as, the other two will become the opponents. Construct a list of the three vehicle objects, and whichever one that the user chose should have the initial 'P' (otherwise, 'C' for car, 'T' for truck, and 'M' for motorcycle). Have a loop that repeats until all three vehicles have reached the finish. On each iteration, display the track and the three vehicles. Then prompt the user to move fast, slow, or to do a special move and call the corresponding method based on what they chose. The opponents should then randomly choose to move: 40% chance to go slow, 30% chance to go fast, and 30% chance to do a special move. If the opponent has run out of energy, they should just choose to move slow. The movement methods need an argument for the location of the next obstacle, you can use the list's index(item, v_pos) method to find the location. If there isn't an obstacle after that position, then it will throw a ValueError that you can catch and set a default value. Place the vehicle's initial at their updated location in the list, place a '*' at their previous location so the user can see the distance that the vehicles have moved. Keep track of first, second, and third place as each vehicle reaches the finish. If the player reaches the finish before one or more opponents, then skip the prompt and play out the rest of the race until all three vehicles have finished and then display the results. Check all input for validity. Add docstrings for all classes and methods.

Example Output

```
Rad Racer!
Choose a vehicle and race it down the track (player = 'P').  Slow down for obstacles ('0') or else
you'll crash!
1. Lightning Car - a fast car. Speed: 7.  Special: Nitro Boost (1.5x speed)
2. Swift Bike - a speedy motorcycle. Speed: 8.  Special: Wheelie (2x speed but there's a chance
you'll wipe out).
3. Behemoth Truck - a heavy truck. Speed: 6.  Special: Ram (2x speed and it smashes through
obstacles).
Choose your vehicle (1-3): 2
Lightning Car [Position - 0, Energy - 100]
Swift Bike [Position - 0, Energy - 100]
Behemoth Truck [Position - 0, Energy - 100]
C---------------------------------0--------0-------------------------------------------------------
P-----------------------------------------------------0------0-------------------------------------
T---------------------------------0-----0-------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 3

Swift Bike pops a wheelie and moves 15 units!!
Lightning Car slowly moves 4 units!
Behemoth Truck slowly moves 3 units!

Lightning Car [Position - 4, Energy - 100]
Swift Bike [Position - 15, Energy - 85]
Behemoth Truck [Position - 3, Energy - 100]
*---C----------------------------0--------0-------------------------------------------------------
*--------------P-----------------------------------------0------0-------------------------------------
*--T----------------------------0-----0-------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 3

Swift Bike pops a wheelie and moves 16 units!!
Lightning Car quickly moves 8 units!
Behemoth Truck quickly moves 5 units!

Lightning Car [Position - 12, Energy - 95]
Swift Bike [Position - 31, Energy - 70]
Behemoth Truck [Position - 8, Energy - 95]
*---*-------C--------------------0--------0-------------------------------------------------------
*--------------*---------------P-------------------------0------0-------------------------------------
*--*----T------------------------0-----0-------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 2

Swift Bike slowly moves 5 units!
Lightning Car uses nitro boost and moves 9 units!
Behemoth Truck quickly moves 6 units!

Lightning Car [Position - 21, Energy - 80]
Swift Bike [Position - 36, Energy - 70]
Behemoth Truck [Position - 14, Energy - 90]
*---*-------*--------C------------0--------0-------------------------------------------------------
*--------------*---------------*----P--------------------0------0-------------------------------------
*--*----*-----T--------------------0-----0-------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 1

Swift Bike quickly moves 7 units!
Lightning Car uses nitro boost and moves 9 units!
Behemoth Truck rams forward 12 units!

Lightning Car [Position - 30, Energy - 65]
Swift Bike [Position - 43, Energy - 65]
Behemoth Truck [Position - 26, Energy - 75]
*---*-------*--------*--------C---0--------0-------------------------------------------------------
*--------------*---------------*----*------P--------------0------0-------------------------------------
*--*----*-----*-----------T-------0-----0-------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 1

Swift Bike quickly moves 8 units!
Lightning Car slowly moves 3 units!
Behemoth Truck quickly moves 5 units!
```

```
Lightning Car [Position - 33, Energy - 65]
Swift Bike [Position - 51, Energy - 60]
Behemoth Truck [Position - 31, Energy - 70]
*---*-------*--------*--------*--C0--------0-------------------------------------------------------------
*--------------*---------------*----*------*-------P-----0------0-----------------------------------
*--*----*-----*-----------*----T--0-----0-------------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 2

Swift Bike slowly dodges the obstacle and moves 7 units!
Lightning Car slowly moves 4 units!
Behemoth Truck slowly moves 3 units!

Lightning Car [Position - 37, Energy - 65]
Swift Bike [Position - 58, Energy - 60]
Behemoth Truck [Position - 34, Energy - 70]
*---*-------*--------*--------*--*0--C-----0-------------------------------------------------------------
*--------------*---------------*----*------*-------*-----0P-----0-----------------------------------
*--*----*-----*-----------*----*--T-----0-------------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 2

Swift Bike slowly dodges the obstacle and moves 7 units!
Lightning Car CRASHED into an obstacle!
Behemoth Truck slowly moves 4 units!

Lightning Car [Position - 43, Energy - 60]
Swift Bike [Position - 65, Energy - 60]
Behemoth Truck [Position - 38, Energy - 70]
*---*-------*--------*--------*--*0--*-----C-------------------------------------------------------------
*--------------*---------------*----*------*-------*-----0*-----0P-----------------------------------
*--*----*-----*-----------*----*--*---T-0-------------------------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 1

Swift Bike quickly moves 9 units!
Lightning Car slowly moves 4 units!
Behemoth Truck rams forward 13 units!

Lightning Car [Position - 47, Energy - 60]
Swift Bike [Position - 74, Energy - 55]
Behemoth Truck [Position - 51, Energy - 55]
*---*-------*--------*--------*--*0--*-----*---C---------------------------------------------------------
*--------------*---------------*----*------*-------*-----0*-----0*--------P-------------------------
*--*----*-----*-----------*----*--*---*-0----------T-----------------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 1

Swift Bike quickly moves 9 units!
Lightning Car uses nitro boost and moves 9 units!
Behemoth Truck quickly moves 6 units!

Lightning Car [Position - 56, Energy - 45]
Swift Bike [Position - 83, Energy - 50]
Behemoth Truck [Position - 57, Energy - 50]
*---*-------*--------*--------*--*0--*-----*---*--------C-----------------------------------------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------P-----------------
*--*----*-----*-----------*----*--*---*-0----------*-----T---------------------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 1

Swift Bike quickly moves 7 units!
Lightning Car slowly moves 3 units!
Behemoth Truck rams forward 13 units!

Lightning Car [Position - 59, Energy - 45]
Swift Bike [Position - 90, Energy - 45]
Behemoth Truck [Position - 70, Energy - 35]
*---*-------*--------*--------*--*0--*-----*---*--------*--C-----------------------------------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------P---------
*--*----*-----*-----------*----*--*---*-0----------*-----*------------T-----------------------------
Choose action (1. Fast, 2. Slow, 3. Special Move): 1

Swift Bike quickly moves 9 units!
Lightning Car slowly moves 3 units!
Behemoth Truck rams forward 12 units!
```

```
Lightning Car [Position - 62, Energy - 45]
Swift Bike [Position - 99, Energy - 40]
Behemoth Truck [Position - 82, Energy - 20]
*---*-------*--------*--------*--*0--*-----*---*--------*--*--C-----------------------------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------T-----------------
Lightning Car slowly moves 3 units!
Behemoth Truck slowly moves 2 units!

Lightning Car [Position - 65, Energy - 45]
Swift Bike [Position - 99, Energy - 40]
Behemoth Truck [Position - 84, Energy - 20]
*---*-------*--------*--------*--*0--*-----*---*--------*--*--*--C-----------------------------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------*-T---------------
Lightning Car uses nitro boost and moves 11 units!
Behemoth Truck rams forward 13 units!

Lightning Car [Position - 76, Energy - 30]
Swift Bike [Position - 99, Energy - 40]
Behemoth Truck [Position - 97, Energy - 5]
*---*-------*--------*--------*--*0--*-----*---*--------*--*--*--*----------C-----------------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------*-*-----------T--
Lightning Car slowly moves 3 units!
Behemoth Truck tries to ram forward, but is all out of energy!

Lightning Car [Position - 79, Energy - 30]
Swift Bike [Position - 99, Energy - 40]
Behemoth Truck [Position - 98, Energy - 5]
*---*-------*--------*--------*--*0--*-----*---*--------*--*--*--*----------*--C--------------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------*-*-----------*T-
Lightning Car slowly moves 4 units!
Behemoth Truck tries to ram forward, but is all out of energy!

Lightning Car [Position - 83, Energy - 30]
Swift Bike [Position - 99, Energy - 40]
Behemoth Truck [Position - 99, Energy - 5]
*---*-------*--------*--------*--*0--*-----*---*--------*--*--*--*----------*--*---C----------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------*-*-----------**T
Lightning Car slowly moves 2 units!

Lightning Car [Position - 85, Energy - 30]
Swift Bike [Position - 99, Energy - 40]
Behemoth Truck [Position - 99, Energy - 5]
*---*-------*--------*--------*--*0--*-----*---*--------*--*--*--*----------*--*---*-C--------------
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------*-*-----------**T
Lightning Car uses nitro boost and moves 11 units!

Lightning Car [Position - 96, Energy - 15]
Swift Bike [Position - 99, Energy - 40]
Behemoth Truck [Position - 99, Energy - 5]
*---*-------*--------*--------*--*0--*-----*---*--------*--*--*--*----------*--*---*-*----------C---
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------*-*-----------**T
Lightning Car slowly moves 4 units!
*---*-------*--------*--------*--*0--*-----*---*--------*--*--*--*----------*--*---*-*----------*--C
*--------------*---------------*----*------*-------*-----0*-----0*--------*--------*------*--------P
*--*----*-----*-----------*----*--*---*-0----------*-----*------------*-----------*-*-----------**T
1st place: Swift Bike [Position - 99, Energy - 40]
2nd place: Behemoth Truck [Position - 99, Energy - 5]
3rd place: Lightning Car [Position - 100, Energy - 15]
```

Notes:

1. You should have 6 different files: vehicle.py, car.py, motorcycle.py, truck.py, main.py, and check_input.py.
2. Check all user input using the get_int_range function in the check_input module.
3. Do not create any extra attributes, methods, or parameters in your classes. You may create functions in main.py as needed.
4. Use decorators to make the abstract method, and properties.
5. Please do not create any global variables or use the attributes globally (ie. do not access any of them using the underscore in the main). You can access the vehicle's initial, energy, and position using the get properties, but you shouldn't access the _speed, or name outside of their own class or subclasses.
6. Use docstrings to document each of the classes, their attributes, and each of their methods. See the lecture notes for examples.
7. Place your names, date, and a brief description of your program in a comment block at the top of your program. Place brief comments throughout your code.
8. Thoroughly test your program before submitting:
    a. Make sure that the fast method checks that the vehicle has sufficient energy and that amount is deducted, it randomizes a value between speed +/- 1, it checks if it will hit an obstacle, and returns the appropriate string.
    b. Make sure that the slow method randomizes a value at 50% speed +/- 1 and half speed (motorcycle is 75% speed), it checks if it will encounter an obstacle, and it returns the appropriate string.
    c. Make sure that the special_move methods for each of the vehicles checks that the vehicle has sufficient energy and that amount is deducted, it randomizes using the correct speeds (car-1.5x, motorcycle-2x(if it didn't fall over), truck-2x), reacts correctly to obstacles (car-crash, motorcycle-crash, truck-rams), and returns the appropriate string.
    d. Make sure that fast and special move methods move 1 space when there isn't sufficient energy.
    e. Make sure that the track is properly set up with 2 obstacles per lane.
    f. Make sure that the track is properly updated with the '*'s and initials as the vehicles move.
    g. Make sure that the vehicles detect and react to the obstacles correctly.
    h. Make sure that after a vehicle reaches the finish line, you do not try to place the initial out of bounds of the list.

Rad Racer Rubric – Time estimate: 5 hours

Rad Racer **10 points**	Correct. 2 points	A minor mistake. 1.5 points	A few mistakes. 1 point	Several mistakes. 0.5 points	No attempt. 0 points
Vehicle class (in a separate file)**:** 1. It is abstract. 2. Has correct attributes and properties. 3. Has correct methods. 4. Has abstract method – no code.					
Car, Motorcycle, Truck classes: 1. Each in separate files. 2. Inherits from Vehicle class. 3. Overrides special_move method. 4. Motorcycle overrides slow method.					
Main file (in a separate file)**:** 1. Creates a 2D list for the track. 2. Track has 3 tracks 100 units long. 3. Track has 2 rand obstacles per lane. 4. Creates a list with 3 vehicle objects. 5. Has a loop that repeats until all 3 vehicles finish the race. 6. Prompts and moves user's vehicle. 7. Randomly chooses oppnt's moves. 8. Finds location of obstacles.					
Output: 1. Track is displayed with obstacles and vehicles' positions. 2. Player's vehicle is labeled with 'P'. 3. Prev positions are replaced with '*'. 4. Vehicles react correctly to obstacles. 5. Displays correct event strings. 6. Displays correct win results.					
Code Formatting: 1. Meaningful variable names. 2. No exceptions thrown. 3. No global variables or accessing attributes directly. 4. Correct documentation.					