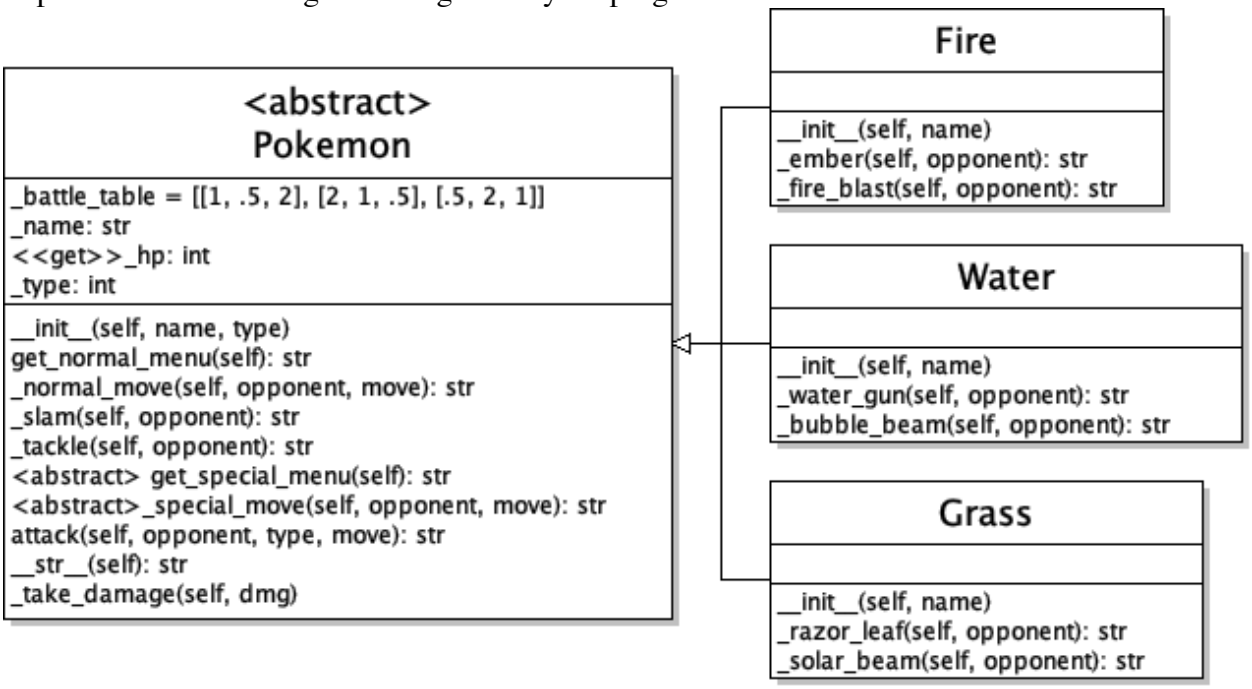
CECS 277 – Lab 8 – Abstract

Pokémon Battle

Create a game where the user must defeat three pokemon to win the game. Use inheritance to implement the following class diagram in your program.



Abstract Pokemon Class (pokemon.py) –

1. __init__(self, name, p_type) – set the _name, and _poke_type using the parameters, assign the _battle_table the 2D list given above, and set _hp to 25.
2. hp property – use a decorator to get (not set) the value of _hp.
3. get_normal_menu(self) – returns a string with the menu options for the normal moves: slam and tackle.
4. _normal_move(self, opponent, move) – use the move parameter to choose to call either slam or tackle method, returns the string returned from those methods.
5. _slam(self, opponent) and _tackle(self, opponent) – randomize some damage (slam 2-6, tackle 3-5), call take_damage on the opponent, and return a string description of the move using both pokemons' names, the name of the move, and the amount of damage taken.
6. get_special_menu(self)->str – abstract (overridden in the subclasses) – returns a string for the menu for the special moves of each pokemon type.
7. _special_move(self, opponent, move)->str – abstract (overridden in the subclasses) – uses the move parameter to choose to call either of the special moves for that pokemon type.
8. attack(self, opponent, type, move) – use the type parameter to choose to call either _normal_move or _special_move and pass it the opponent and move parameters.
9. __str__(self) – display the pokemon's name and hp in the format "Name: hp/25".
10. _take_damage(self, dmg) – the damage the pokemon takes. Subtract the dmg value from the pokemon's _hp. Check that the _hp doesn't go past 0 (if it's negative, reset it to 0).

Fire/Water/Grass Classes (fire.py/water.py/grass.py) – each inherit from Pokemon.

1. __init__(self, name=None) – call super to set the name and type (fire=0, water=1, grass=2). If a name is not provided choose a random one from the lists below.
2. Override the get_special_menu and _special_move methods for this class.
3. Create the two move methods – each pokemon type has two moves:

	Special Move 1	Special Move 2
Fire (Ponyta, Growlithe, Vulpix)	ember: 2-6 dmg	fire blast: 1-7 dmg
Water (Staryu, Magikarp, Horsea)	water gun: 1-7 dmg	bubble beam: 3-5 dmg
Grass (Oddish, Bellsprout, Exeggcute)	razor leaf: 3-5 dmg	solar beam: 2-6 dmg

Randomize damage according to the table above, look up the multiplier in the battle table based on the pokemons' types and calculate the total integer damage the opposing pokemon will take. Then return a string description of the attack using both pokemons' names, the name of the move, the amount of damage taken, and whether it was effective (x2) or not (x.5).

Main (main.py) – Create a list of three pokemon objects, each chosen randomly (ie. fire, water, grass), for the opposing trainer, and display them to the user. Prompt the user to choose a single pokemon that they will battle with, either fire, water, or grass. Every round, allow the user to choose what type of attack to do (ie. normal or special), and then display the menu to allow them to choose which move to do of that type (ie. if the user chose normal, then they can either choose slam or tackle), call the pokemon's attack method using their choices and display the resulting string. If the opposing trainer's pokemon still has hit points, then it will choose a random move to attack back with. When one of the opposing trainer's pokemon is defeated, remove it from the list. Repeat until the user has defeated all three of the opposing trainer's pokemon, or until the user's pokemon is defeated. Display a message when the user wins/loses. Check all input for validity. Document all classes, methods, and functions.

Example Output (user input is in italics)**:**

```
PROF OAK:  Hello Trainer!  Today
you're off to fight your first
battle of 1 vs. 3 pokemon.
1. Horsea HP: 25/25
2. Magikarp HP: 25/25
3. Exeggcute HP: 25/25
Select the pokemon that you will
battle with.
1. I choose you, Charmander.
2. Squirtle!  GO!
3. We can do it together,
Bulbasaur!
Please choose a pokemon: 3

  -- TRAINER BATTLE --
TRAINER: I choose you:
Horsea HP: 25/25

Bulbasaur HP: 25/25
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 2

Choose a Move:
1. Razor Leaf
2. Solar Beam
Enter move: 1

Bulbasaur slices Horsea with RAZOR
sharp LEAVES for 10 damage.  It was
SUPER EFFECTIVE!
Horsea blasts Bulbasaur with a
WATER GUN for 3 damage.  It was not
very effective.

TRAINER: I choose you:
Horsea HP: 15/25

Bulbasaur HP: 22/25
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 2

Choose a Move:
```

```
1. Razor Leaf
2. Solar Beam
Enter move: 1

Bulbasaur slices Horsea with RAZOR
sharp LEAVES for 8 damage.  It was
SUPER EFFECTIVE!
Horsea SLAMS Bulbasaur for 3
damage.

TRAINER: I choose you:
Horsea HP: 7/25

Bulbasaur HP: 19/25
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 2

Choose a Move:
1. Razor Leaf
2. Solar Beam
Enter move: 2

Bulbasaur blasts Horsea with a
SOLAR BEAM for 8 damage.  It was
SUPER EFFECTIVE!
TRAINER: NOOOOO!  You defeated my
pokemon!

TRAINER: I choose you:
Magikarp HP: 25/25

Bulbasaur HP: 19/25
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 2

Choose a Move:
1. Razor Leaf
2. Solar Beam
Enter move: 2

Bulbasaur blasts Magikarp with a
SOLAR BEAM for 12 damage.  It was
SUPER EFFECTIVE!
Magikarp SLAMS Bulbasaur for 6
damage.

TRAINER: I choose you:
Magikarp HP: 13/25

Bulbasaur HP: 13/25
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 2

Choose a Move:
1. Razor Leaf
2. Solar Beam
Enter move: 1

Bulbasaur slices Magikarp with
RAZOR sharp LEAVES for 8 damage.
It was SUPER EFFECTIVE!
Magikarp blasts Bulbasaur with a
WATER GUN for 1 damage.  It was not
very effective.

TRAINER: I choose you:
Magikarp HP: 5/25

Bulbasaur HP: 12/25
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 2

Choose a Move:
1. Razor Leaf
2. Solar Beam
Enter move: 1

Bulbasaur slices Magikarp with
RAZOR sharp LEAVES for 8 damage.
It was SUPER EFFECTIVE!
TRAINER: NOOOOO!  You defeated my
pokemon!

TRAINER: I choose you:
Exeggcute HP: 25/25

Bulbasaur HP: 12/25
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 1

Choose a Move:
1. Slam
2. Tackle
Enter move: 2

Bulbasaur TACKLES Exeggcute for 4
damage.
Exeggcute blasts Bulbasaur with a
SOLAR BEAM for 2 damage.

TRAINER: I choose you:
Exeggcute HP: 21/25

Bulbasaur HP: 10/25
```

```
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 1

Choose a Move:
1. Slam
2. Tackle
Enter move: 2

Bulbasaur TACKLES Exeggcute for 3
damage.
Exeggcute SLAMS Bulbasaur for 5
damage.

TRAINER: I choose you:
Exeggcute HP: 18/25

Bulbasaur HP: 5/25
```

```
Choose an Attack Type:
1. Normal
2. Special
Enter attack type: 1

Choose a Move:
1. Slam
2. Tackle
Enter move: 2

Bulbasaur TACKLES Exeggcute for 3
damage.
Exeggcute blasts Bulbasaur with a
SOLAR BEAM for 5 damage.

TRAINER: HA!  I defeated you, come
back when you get a better
pokemon...
```

Notes:

1. You should have 5 different files: pokemon.py, fire.py, water.py, grass.py, and main.py.
2. Use docstrings to document each of the classes, their attributes, and each of their methods. See the lecture notes for examples.
3. Place your names, date, and a brief description of your program in a comment block at the top of your main file. Place brief comments throughout your code.
4. Do not create extra attributes, methods, or parameters (you may create functions in main).
5. Please do not create any global variables or use the attributes globally. Use the methods instead. You can use the attributes in the subclasses when needed. You can access the hp in main using the property. Only use the public methods (non-underscored) in main.
6. Check all user input using the get_int_range function in the check_input module.
7. Feel free to add text to make the game more interesting and to add flair. You may also modify the random amounts of damage to better balance the game.
8. List of pokemon names and types: https://pokemondb.net/pokedex/game/red-blue-yellow.
9. The attribute 'poke_type' and the parameter 'type' in the attack method are not the same.
10. The battle table uses the pokemon's type values: attacker = row, and defender = column.
11. Thoroughly test your program before submitting:
    a. Make sure that your classes are inherited properly. Fire, Water, and Grass should inherit from Pokemon, and Pokemon should be abstract with two abstract methods.
    b. Make sure user input is validated.
    c. Make sure that you are calling the appropriate menu methods to display the menus.
    d. Make sure that each type of pokemon does the proper special attacks.
    e. Make sure that the normal attacks do not use the multiplier from the battle table.
    f. Make sure that the special attacks do use the multiplier from the battle table.
    g. Make sure that the correct multiplier is used for the two types of pokemon (ex. Fire is super effective against Grass and does double damage).
    h. Make sure that the damage dealt is correctly subtracted from the opponent.
    i. Make sure that the defeated pokemon are removed from the opponent trainer's list.
    j. Make sure the game ends when the user defeats all 3 pokemon, or when the user's pokemon runs out of hp.

Pokemon Battle Rubric – Time estimate: 4 hours

Pokemon Battle **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
Pokemon class (in a separate file): 1. Is abstract with 2 abstract methods. 2. Has 4 attributes and 5 methods with a leading underscore (plus init and str). 3. Has property for hp. 4. take_dmg ensures that hp > 0.					
Fire/Water/Grass classes (sep files): 1. All inherit from Pokemon class. 2. init chooses a random name if not provided and uses super to set poke_type to F=0,W=1, G=2.					
Menu and attack methods: 1. normal_move calls slam/tackle and returns attack string. 2. slam and tackle deal damage and return the attack string. 3. special menus return correct menu string for each type. 4. special moves call correct methods for each type and return attack string. 5. special moves use battle table. 6. attack method calls normal or special move and returns attack string.					
Main File: 1. Creates list of 3 opponent pokemon. 2. Creates user's chosen pokemon. 3. Repeatedly prompts user for attack type and move type to perform attack. 4. Opponent randomly chooses attack and move type to attack back (if hp>0). 5. Calls attack not the move methods. 6. Repeats until user hp=0 or there are no more opponents left. 7. Error checks all user input. 8. Does not use any underscored methods or attributes in main.					
Code Formatting: 1. All code is in functions/methods. 2. Correct spacing and good naming. 3. No exceptions thrown. 4. No global variables. 5. Correct documentation.					