

## EN.540.635 “Software Carpentry”

### Lab 8 - Pokemon Index

Today, we’ll practice using classes as we code up a simple Pokemon game. You are a trainer with both a pokedex and a pokebag. The world is infinitely large with no roadblocks, and you have the following options available to you:

```
Go north , Go east , Go , Print pokedex , Print Pokebag , Get pkmn from pokedex ,  
Get pkmn from pokebag , exit , quit , stop
```

Your goal is to catch all the Pokemon in this world (gotta catch ‘em all, right?). As you explore, you will randomly encounter Pokemon. If you encounter a Pokemon that you haven’t seen before, it will be added to your Pokedex. You then have the option of catching the Pokemon, which I may say, isn’t always successful.

With each step you walk, all Pokemon that you have in your bag will level up (read: rare candy *everywhere*). In fact, all their stats will increase by 1. But... Pokemon don’t evolve in this world, so yes, you’ll have to catch all the Pokemon in a single evolution line. Once you’ve caught all the Pokemon (9 in this game), you’ll get a “Congratulations!” message. Of course, you can keep playing the game.

We’ve already built in the “game” functionality, but your job is to help us create the Pokemon, Pokedex, and Pokebag classes such that the game functions.

A pokemon has the following properties:

```
Index , Name , Type(s) , Move(s) ,  
Stats ( Dictionary of Level , HP , Attack , Defense , Sp. Atk , Sp. Def , Speed )
```

Note that there are a few differences between this Pokemon game and the one we are used to:

- Pokemon do not evolve (yet!). We are not coding up the evolution in this lab; but think, how could you do so? Inheritance, maybe?
- Your bag is infinite. If it was finite, what would you do?
- There are no roadblocks. If you had to program in an actual “world,” how would you do so?