CIS 211, S20   Lab 2   Name ____________________

Overview

1. **Review of last week** - what is a class, what is an object, and what does self mean?
2. **Methods in same class** - how to call a method in same class?
3. **Inheritance** - what is a subclass and how to make one?
4. **Methods in super class** - how to (specifically) call methods of a super class from a subclass?
5. **Overriding** - how to customize methods for subclasses?
6. **Dynamic Dispatch** - what does that mean and how to take advantage of it?

Exercise

1 Make a `Python` file with whatever name you like. Create a class `Phone`, whose constructor takes a parameter `name` of type `str`. Inside the constructor, create a class variable to save `name`.

2 Make a method `pickUp` in class `Phone` that prints `Picking up <name of Phone>`.

3 Copy the following code into your `Phone` class. (It helps more to type it yourself! Also be careful about indentation.)

```
    def dial(self, num: str):
        print(f"dialing {num}")

    def speak(self, msg: str):
        print(f"speaking: {msg}")
```

4 Make a method with signature `call(self, num: str, msg: str)`, which calls methods `pickUp`, `dial`, and `speak`. (Now you should be able to answer **Overview** question 2.)

5 Make a subclass `SmartPhone` of class `Phone` by starting with the following code. `SmartPhone` takes the same parameter as its superclass `Phone` does and calls `__init__` method of its superclass. (Now you should be able to answer **Overview** question 3 and 4.)

6 Copy the following method into class `SmartPhone`.

```
    def openPhoneApp(self):
        print(f"opening Phone App on {self.name}")
```

7 Make a `call` method with same method signature as in step 4 in class `SmartPhone`. This time, call methods `openPhoneApp`, `dial`, and `speark`. (Now you know how to override a method and answer **Overview** quesiont 5.)

8 Copy the following function into your file and answer the questions below. (I'm using the term 'function' to address that it does not go inside any class, and the fact that it does not take a `self` makes it more obvious.)

```
def main():
    ph1 = Phone("Panasonic")
    ph2 = Phone("AT&T")
    sPh1 = SmartPhone("iPhone")
    sPh2 = SmartPhone("Google")
    phoneList = [ph1, ph2, sPh1, sPh2]
    for phone in phoneList:
        phone.call("5413461000", "This is CIS 211 Lab.")
        print("________________________________________")
```

What does each line of the function do? Which lines are creating `Phone` objects and `SmartPhone` objects? What does the loop do? What should the function print?

9 Call the `main` method in `if __name__ == ’__main__’:` and run your code. What does it print and why? (Now you have answer to **Overview** question 6.)