

## Programming Assignment 4

Due Wednesday, November 30 at 11:59 pm

Create a Person Class and a Student class that inherits from it. All input data for both classes is as strings. Expected methods and their behaviors are as follows:

Class **Person**: firstName, lastName and SSN must be provided. Also to be initialized: **phones** starts as an empty list and **address** starts as "<no address>"

Methods that must be included:

**\_\_init\_\_** - Requires **firstName**, **LastName**, **SSN** and nothing else.

**\_\_str\_\_** -see outputs below.

**printAll** -see outputs below. Prints "<none>" if no phone is currently specified.

The actions for the following should be obvious from the name or by looking at the sample code below:

**getFirstName**, **setFirstName**, **getLastName**, **setLastName**, **getSSN**, **getAddress**, **setAddress**, **addPhone**, **delPhone**, **getPhones**. There is no setSSN method.

Class **Student**: Must inherit from Person. Also needs **techID** and **grades** which is initialized as an empty dictionary which will have courses as keys and their grades as the associated values.

**printAll** – see outputs below. This is different than the **printAll** in **Person**. Prints "<none>" if no phone has been specified and also if no courses are in the **grades** dictionary.

**addCourse** – add a course ID and it's associated grade value to the **grades** dictionary. Also used to change a course grade. Grades are always stored and shown in upper case.

**dropCourse** – remove a course from the **grades** dictionary. Dropping non-existent courses does not cause an error. A course can only be dropped if its grade begins with 'I'

I will run your file to see what your test code does and will also import the classes (**which must be named Person and Student**) to exercise it with my test code. Submit to D2L as PA4.py. As usual, extra credit applies for submitting early.

Sample testing code and output that it should produce are on the next pages. This testing code is also on D2L. Note that not all of the specifications above are tested by the following code but you should implement and test these features.

```

if __name__ == "__main__":
    per1 = Person("Joe", "Smith", "111-22-3333")
    per2 = Person("Lisa", "Hernandez", "222-33-4444")
    per2.setAddress("123 South St #456, Chicago, IL, 61111")
    per2.addPhone("312-234-5678")
    per2.addPhone("800-888-9990")

    print(per1)
    print(per2)
    print()
    per1.printAll()
    print()
    per2.printAll()
    print()
    print("Result of getPhones():", per2.getPhones())
    print()

    stu1 = Student("Mary", "Jones", "234-56-1234", "00112233")
    stu1.setAddress("333 N Main St, Dallas, TX 75150")
    stu2 = Student("Bill", "Chen", "876-76-7676", "12345433")
    stu2.addPhone("214-999-0011")
    stu2.addCourse("IT 210", "IP")
    stu2.addCourse("ENG 271W", "IP")
    stu2.addCourse("MATH 121", "IP")

    print(stu1)
    print(stu2)
    print()
    stu1.printAll()
    print()
    stu2.printAll()

    stu2.addCourse("IT 210", "a") # Change grade
    stu2.dropCourse("ENG 271W") # Drop existing course
    stu2.dropCourse("ENG 101") # Dropping a non-existent course does nothing
    stu2.addPhone("214-999-5511")
    stu2.delPhone("214-999-0011") # Delete existing phone
    stu2.delPhone("999-999-9999") # Delete non-existent phone does nothing

    print()
    stu2.printAll()

```

## Output:

Smith, Joe SSN=111-22-3333  
Hernandez, Lisa SSN=222-33-4444

Smith, Joe SSN=111-22-3333  
<no address>  
Phone(s):  
<none>

Hernandez, Lisa SSN=222-33-4444  
123 South St #456, Chicago, IL, 61111  
Phone(s):  
312-234-5678  
800-888-9990

Result of getPhones(): ['312-234-5678', '800-888-9990']

Jones, Mary techID=00112233  
Chen, Bill techID=12345433

Jones, Mary techID=00112233  
333 N Main St, Dallas, TX 75150  
Phone(s):  
<none>  
Courses:  
<none>

Chen, Bill techID=12345433  
<no address>  
Phone(s):  
214-999-0011  
Courses:  
ENG 271W IP  
IT 210 IP  
MATH 121 IP

Chen, Bill techID=12345433  
<no address>  
Phone(s):  
214-999-5511  
Courses:  
IT 210 A  
MATH 121 IP