CSC 4992
Section 002
Special Topics in Computer Science
Python Programming
Winter Term 2018
Assignment 04
50 points
Due (04/05/2018, 11:55 P.M.)

Submit a source file (.py) for each question.

Problem 1 (25 points)
Suppose class Student represents information about students in a course. Each student has a name and a list of test scores. The Student class should allow the user to view a student's name, view a test score at a given position, view the highest test score, view the average test score, and obtain a string representation of the student's information. When a Student object is created, the user supplies the student's name and the number of test scores. Each score is initially presumed to be 0. Complete the missing codes in the following program.

```
class Student(object):
   """Represents a student."""

   def __init__(self, name, number):
      """All scores are initially 0."""
      self._name = name
      self._scores = []
      for count in xrange(number):
         self._scores.append(0)

   def getName(self):
      """Returns the student's name."""
      #your code here

   def setScore(self, i, score):
      """Resets the ith score, counting from 1."""
      #your code here

   def getScore(self, i):
      """Returns the ith score, counting from 1."""
      #your code here

   def getAverage(self):
      """Returns the average score."""
```

```
        #your code here

    def getHighScore(self):
        """Returns the highest score."""
        #your code here

    def __str__(self):
        """Returns the string representation of the student."""
        #your code here
```

Problem 2 (15 points)
Define a class for a restricted saving account that only permits three withdraw per months (see SavingsAccount class in slides)

```
class RestrictedSavingsAccount(SavingsAccount):
    """This class represents a restricted savings account."""

    MAX_WITHDRAWALS = 3

    def __init__(self, name, pin, balance = 0.0):
        """Same attributes as SavingsAccount, but with
        a counter for withdrawals."""

    def withdraw(self, amount):
        """Restricts number of withdrawals to MAX_WITHDRAWALS."""

    def resetCounter(self):
        self._counter = 0
```

Problem 3 (10 points)

Suppose class Person is the parent of class Employee. Complete the following code:

```
class Person:
    def __init__(self, first, last):
        self.firstname = first
        self.lastname = last
    def Name(self):
        return self.firstname + " " + self.lastname


class Employee(Person):
    def __init__(self, first, last, staffnum):
    """Intialzie firstnames and lastname, staffnumber """
      # write your code here
    def GetEmployee(self):
     """Return Employee's Name(firstnames and lastname) and
staffnumber"""
       # write your code here
x = Person("Marge", "Simpson")
y = Employee("Homer", "Simpson", "1007")
print(x.Name())
print(y.GetEmployee())
```