8.6 Exercises

Write a complete Python program to solve each of the following problems.

8.1. Make a program containing class Student. This class should have two instance variables: name and age. The constructor takes a name argument and uses it to initialize the name instance variable; it also initializes the age instance variable to 18. The class has a SetAge() method that can set the age to any other value. Finally, the class has a Print() method. Here is a main() method for your program:

```
def main():
    x = Student( "bob" )
    x.SetAge(23)
    x.Print()
```

8.2. Give class Student in problem 8.1 an instance variable Gender, which can be "M" or "F". Modfiy the constructor for Student to take a gender argument as well as a name argument to give a value to this variable.

8.3. Give the Student class in problem 8.2another instance variable: transcript . This is a list of pairs, where the first element of the pair is the name of a course, and the second element is a numeric grade between 0 and 4.There are two additional methods:

AddCourse(self) which asks the user for a course name and grade, and appends this pair onto the transcript, and

GPA(self) which averages all of the grades in the transcript and returns the average.

The main program

```
def main():
    x = Student( "bob", "M")
    x.AddCourse( )
    x.AddCourse()
    print X.GPA()
```

will cause the system to twice ask for course name and grade. If you supply the following information:

Course name? `Basketweaving`
Grade for Basketweaving? `4`
Course name? `Time Wasting`
Grade for Time Wasting? `3`

then the print statement **print** X.GPA at the end of main() will print 3.5

8.4. Give class Student in problem 8.1 another instance variable Roommate. This can either have value None, or an object of class Student. Give the class a method AssignRoommate() to give a value to this variable.

8.5. Make another class Faculty in problem **??**, which just has one instance variable, varname. The only methods Faculty needs are a constructor and a Print() method. Add to class Student an instance variable advisor. This should be initialized to None in the Student constructor. Give Student a method AssignAdvisor() that assigns an advisor (a faculty member) to the student.

8.6. The relationship between Faculty and Student in problem 8.5 is too one-sided. Give to class Faculty an instance variable Advisees, which is a list of all of the advisees the faculty member has. Add to Faculty a method AssignAdvisee() that will append a student to the Advisees list. If x is a Student and y is a Faculty, then x.AssignAdvisor(y) and y.AssignAdvisee(x) should do the same thing.