Seminar 1 (Classes and Objects) - Exercises

1. a. Write a class Name that models a person's name.
   The class has 4 instance variables – gender (str), last name (str), first name(str), middle name(str).

   It has a constructor that initializes the gender, last, first and middle name. It has getter and setter methods only for first name, middle name and last name.

   It has the following methods:
   - getFullName() returns the full name with a salutation "Mr." or "Ms.", depending on the person's gender (m or f). The name is given in this orderL the last name, first name followed by the middle name, e.g., Mr. Ong Ah Kow"
   - getInitials() returns the first letter of the first, middle followed by the lastname. E.g. it may return "A. K. Ong".
   - __str__() method that returns a string representation of a Name object

   b. Write a main method to create a Name object and test out the methods in the class.

2. a. Write a class that models a Rectangle.
   A Rectangle class has 2 instance variables – length (float) and width (float).

   It has a constructor that initializes the length and width. It also has get/set methods for the length and width.

   It has the following methods:
   - getArea() that returns the area of the rectangle
   - getPerimeter() that returns the perimeter of the rectangle
   - increaseSize(length, width) that increases the length and width of the rectangle by the given amounts.
   - isBigger(r) that returns a boolean value. The method returns True if the current area is bigger than the area of the rectangle r, passed as parameter. It returns False otherwise.
   - __str__() method that returns a string representation of a Rectangle object, including its area and perimeter.

   b. Write a main method to create a Rectangle object and test out the methods in the class.

3. a. Write a class Point that models a 2 dimensional point (x,y)
   The Point class has 2 instance variables, x and y.

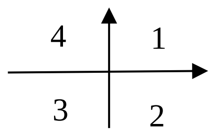
   It has a constructor that initializes 2 instance variables with the value of x and y, with default values (0,0).

   It has the following methods:
   - Getter and setter properties for x and y .

- A move(dx, dy) method that moves to (x+dx, y+dy).
- A distanceTo(aPoint) method that returns the distance to another point (x1, y1). The distance is calculated by this formula:

$$distance = \sqrt{(x - x1)^2 + (y - y1)^2}$$

- A quadrant() method that returns the quadrant of the point as follows:



  For any point along x or y axis, return 0.
- A __str__() method that returns the string value in this format: (x, y)

b. Test the Point class, with the following:

- Create a point object p1, at (5, 1)
- Print the coordinates of p1
- Move p1 by delta (5, -5)
- Create another point p2 at (10, -10)
- Print the distance between p1 and p2
- Print the quadrants for p1 and p2

4. a. Write a class to model a BankAccount.
   A BankAccount class has 3 instance variables: accountId, pin and balance.

   It has a constructor that has 3 parameters to initialize the accountId, pin and the balance. The default balance is $100. It has getter properties for accountId pin, and balance. It has setter properties for pin and balance.

   It has the following methods:
   - A changePin method that has old pin and new pin as parameters. The new pin is updated only if the old pin matches the existing pin. Return true if the change is successful and false otherwise.
   - A deposit method that accepts an argument that represents the amount to deposit. The method adds the amount to the balance.
   - A withdraw method that accepts an argument which represents the withdrawal amount. This method returns a boolean value. If the withdrawal amount is greater than the balance, no withdrawal is made and a false value is returned. Otherwise, the amount is deducted from the balance and a true value is returned.
   - A transfer method that accepts 2 arguments – another bank account to transfer to and the amount to transfer.
   - The __str__() method returns the accountId and balance as a string.

   b. Write the BankAccount class and write a main method for the following:
   - Declare a BankAccount object for id='B1', pin=111
   - Make a deposit of $100 for id='B1'. Display the status.
   - Change the pin for id='B1'.
   - Declare another BankAccount object for id='B2'

- Make withdrawal amount of $200 for id='B2' and display whether the withdrawal is successful.
- Transfer $100 from bank account 'B1' to 'B2'. Display whether the transfer is successful, and if so, the bank balances of both accounts.
- Display the total amount of all the BankAccount objects in the list.

5. a. Write a Phone class that encapsulates speed dialing feature of a phone. Speed dialing allows one of the digits (0-9) to be assigned a phone number so that the digit can be used to quickly dial a number. The Phone class consists of the following:
   - Instance variables:
     - the phone number of itself (int).
     - a collection (empty list) to store numbers for speed dialing.

   - Constructor
     The constructor has a parameter that initializes its phone number and the collection.

   - It has the following methods:
     - assignSpeedDial(digit, phoneNumber)
       This method assigns a digit with a number. Display "invalid digit" if the digit is not between 0-9.
     - speedDial(digit)
       This method dials a number based on the digit parameter (must be 0-9). If the digit has been assigned with a number (i.e. not zero), then display "calling 999999", where 999999 is the number assigned to the digit. If the digit is not assigned with a number, display "No number assigned".
     - displayAllSpeedDial()
       This method displays the numbers assign to each digit(0-9). A sample is as follows:

```
0 – not assigned
1 – 91232456
2 – 81234567
3 – not assigned
...
9 – 82468124
```

   b. Test the Phone class.

6. a. Write a ToDo class that allows a user to record things to do for an event, e.g. an upcoming overseas trip.

   The class has 2 instance parameters. They are:
   - the event (string)
   - a list collection to store the to-do actions related to the event

   It has getter property for the event name.

The constructor that has an event as parameter. The constructor initializes an empty collection.

It has the following methods:

- A addToDo(toDo) method that adds the toDo (str) action to the collection.
- A displayToDoList() method that displays all the toDo action items in the following format:

  Event: travelling
  1. Bring passport
  2. Change money
  3. Bring medicine

  …
- A deleteToDoItem(index) method that removes a to-do item using the index as shown in part d). So deleteToDoItem(2) removes the ‘change money’ item.

b. Test the ToDo class, with the following:
   - Create a ToDo object for a “Things to bring tomorrow” event.
   - Add a few to do actions to the object.
   - Display the to do list.
   - Remove a to do action

7. a. Write a FileAnalyzer class that provides information about the contents of a file.

The class has a instance parameter _contents (string) that stores the contents of the file.

It has a constructor that has a filename as parameter. The constructor opens the file and reads the contents. The contents is stored with an instance variable _contents as a string.

It has the following methods:
- A method displayContents() that merely prints out the contents.
- A method numberOfWords(). The method uses the split() function on the contents. This results in a list and the length of the list is the number of words. Return this number.
- A method displayWithLineNumber() that displays every line with a line number. Split the string using .split(‘\n’), then go through the list and display each line with a line number in front.
- A method search(word) that has a word as parameter. The method lists all lines with line numbers that contains the word.
- A method appendALine(line) that appends a line at the end of the contents.

b. Test the File Analyzer class.

- Create a FileAnalyzer object with your current python program source file.
- Display the contents of the file.
- Display the number of words.
- Display the lines with line number numbers.
- Prompt for a word. Display the lines that contain the word.
- Append a line to the contents and print the contents to verify.