COMP1551 Core Programming

Objects Worksheet (Week 8 - w/c 7th November 2014)

November 17, 2014

Aims:

→ to us Python object syntax
→ to be able to instantiate objects in Python
→ to understand the terminology used in object oriented programming

1. (a) Implement a class named `Rectangle` to represent a rectangle. The class contains:

   - Two instance variables named `width` and `height`.
   - A constructor that constructs a `Rectangle`, the constructor should accept two parameters; width and height. The default values for width and height should be 0.
   - A method named `getArea` that returns the area of the rectangle represented by the object.
   - A method named `getPerimeter` that returns the perimeter of the rectangle represented by the object.
   - A method named `__str__` that returns a string in a format similar to "Width: 8 Height: 9".

   (b) Draw a UML diagram for the `Rectangle` class.

   (c) Write a test program that instantiates two `Rectangle` objects- one with width 4 and height 40 and the other with width 3.5 and height 35.7. For each rectangle print the `Rectangle` object, the perimeter and the area.

   (d) Extend the test program to graphically display the rectangles using tkinter.

2. A regular polygon is a shape where all angles are the same and all sides are straight and are of the same length.

   (a) Implement a class named `RegularPolygon` to represent a regular polygon. The class contains:

   - An integer instance variable named `n` that defines the number of sides the polygon has.
   - A float instance variable named `side` that stores the length of a side of the polygon.

- A float instance variable named x that defines the x-coordinate of the centre of the polygon, the default value should be 0.
- A float instance variable named y that defines the y-coordinate of the centre of the polygon, the default value should be 0.
- A constructor that constructs a RegularPolygon, the constructor should accept 4 parameters; n, side,x, and y with default values 3,1,0 and 0 respectively.
- The accessor and mutator methods for all instance variables (sometimes called getters and setters).
- A method named getArea that returns the area of the polygon represented by the object.
- A method named getPerimeter that returns the perimeter of the polygon represented by the object.
- A method named __str__ that returns a string in a format similar to "Number of sides: 2 Length of side: 4".

(b) Draw a UML diagram for the RegularPolygon class.

(c) Write a test program that instantiates three RegularPolygon objects- one with default parameters, one with 6 sides of length 4 and one with 10 sides of length 4 centered at (5.6,7.8). The program should print each polygon, the perimeter and the area.

(d) Extend the test program to graphically display the polygons using tkinter.

3. (a) Implement a class named QuadraticEquation to represent a quadratic equation with the following coefficients; $ax^2 + bx + c = 0$. The class contains:

- The instance variables a, b and c to represents the coefficients a, b and c respectively.
- A constructor to construct a QuadraticEquation, the constructor should accept 3 parameters a, b and c.
- The accessor methods for each of the coefficients.
- A method named getDiscriminant that returns the discriminant. The discriminant of a quadratic equation is defined as $b^2 - 4ac$.
- The methods getRoot1 and getRoot2 that return the two roots of the quadratic equation represented by the object. The roots of a quadratic equation can be calculated using the following formulae;

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

- A method named __str__ that returns a string in a format similar to "ax^2 + bx + c".

(b) Draw a UML diagram for the QuadraticEquation class.

(c) Write a test program that prompts the user to enter the three coefficients and displays the result based on the discriminant. If the discriminant is;

- positive, then there are two distinct roots. Print both of them.
- 0, then there is a unique root. Print the unique root.
- negative, then there are no real roots. Print "There are no real roots".

(d) Extend the test program to graphically display the curve using tkinter.

4. (a) Implement a class named `LinearEquation` for a 2 × 2 system of the linear equations. For a system of linear equations;

$$ax + by = e \qquad cx + dy = f$$

$$x = \frac{ed - bf}{ad - bc} \qquad y = \frac{af - ec}{ad - bc}$$

The class contains:

- The instance variables `a`,`b`,`c`,`d`, `e` and `f`.
- The accessor methods for each instance variable.
- A constructor to construct a `LinearEquation`, the constructor should accept the parameters `a`,`b`,`c`,`d`, `e` and `f`.
- A method named `isSolvable` that returns `True` if and only if $ad - bc$ is not 0.
- The methods named `getX` and `getY` that return the solution for the equations.

(b) Draw the UML diagram for the `LinearEquation` class.

(c) Write a test program that prompts the user to enter `a`,`b`,`c`,`d`, `e` and `f` and display the solution. if $ad - bc = 0$ then print "The equation has no unique solution".

(d) Extend the test program to draw the lines and label their intersection using tkinter.

Extended Question

A book lending library can be modelled using objects. A library contains a collection of books; each book has a title, a publishing year and an author. An author has a firstname, lastname, a year of birth and a place of birth.

5. (a) Implement the following classes.

- A class named `Author`; the class contains:
  - Instance variables to store the firstname, lastname, year of birth and place of birth.
  - A constructor that constructs an `Author`, the constructor should accept 4 parameters- firstname, lastname, year of birth and place of birth.
  - The accessor methods for each instance variable.
  - A method named `__str__` that returns a string in a similar format to "Rudyard Kippling, 1865, India".

- A class named `Book`; the class contains:
  - Instance variables to store the title, year of publication and author.
  - A constructor that constructs a `Book`, the constructor should accept 3 parameters- title, year of publication and an author object.
  - The accessor methods for each instance variable.
  - A method named `__str__` that returns a string in a similar format to "Just so stories, 1902, Rudyard Kippling".
- A class named `Library`; the class contains:
  - An instance variable to store a list of books in the library.
  - An instance variable to store a list of authors of the books in the library.
  - A method to return a list of books from an author, provided as a parameter to the method.
  - A method to return a list of books published before a date provided as a parameter to the method.
  - A method to add a new book to the library provided as a parameter to the method.
  - A method to remove a book from the library provided as a parameter to the method.
  - A method to add a new author to the library provided as a parameter to the method.
  - A method to save the library to a file, the filename is provided as a parameter to the method. The file should contain the list of books in the library, one book per line.
  - A constructor to construct a `library`.
- Draw a UML diagram to represent the system described.
- Write a test program that provides the following functionality via a terminal based application;
  - A welcome menu, including a list of available operations.
  - A facility to view all books in the library.
  - A facility to view all books published before a user provided date.
  - A facility to view all books by an author, the user should be prompted to select an author from a list.
  - A facility to view all authors, the list should not contain duplicates.
  - A facility to add a new book to the library, the user should be prompted to enter the details of the book and should be prompted to select an author from a list.
  - A facility to add new authors to the library, the user should be prompted to enter the details of the author.
  - A facility to save the library to a file, the user should be prompted for a filename.

Questions adapted from Y Daniel Liang (2013). Introduction to programming using Python. Prentice Hall US: Pearson Education, Inc..