**Introduction to Computer Science II**

**Lab and Homework 2**

**Due by 1:30pm on Tuesday, January 21**

**Reading**

Read Chapter 7 and sections 8.1 and 8.2 of the textbook.

**Problems**

**Implement the below functions in**[homework2.py](http://reed.cs.depaul.edu/lperkovic/csc242/homeworks/homework2.py)**and submit through COL.**  
**0.**    Lab attendance is required and constitutes 20% of the lab grade.  
 **1.**    Develop a class Animal that supports 3 methods: 

* setSpecies(species): Sets the species of the animal object to species.
* setLanguage(language): Sets the language of the animal object to language.
* speak(): Prints a message from the animal as shown below.

Usage:  
>>> snoopy = Animal()  
>>> snoopy.setpecies('dog')   
>>> snoopy.setLanguage('bark')  
>>> snoopy.speak()  
I am a dog and I bark.

**2.**    Implement a class Polygon that represents regular polygons and supports class methods:

* numSides(n): sets the number of sides of the regular polygon object.
* sideLength(s): sets the side length of the regular polygon object.
* perimeter(): returns the perimeter of polygon object.
* area(): returns the area of the polygon object.

Note: the area of a regular polygon with n sides of length s is   (s2\*n) /  (4\*tan(Pi / n))   where tan() is the tangent trig function.

Usage:  
>>> p = Polygon()  
>>> p.numSides(6)  
>>> p.sideLength(1)  
>>> p.perimeter()  
6  
>>> p.area()  
2.5980762113533165

**3.**   Develop a class Craps that allows you to play craps on your computer. Craps is a dice-based game played in many casinos. Like blackjack, a player plays against the house. The game starts with the player throwing a**pair** of standard, six-sided dice. If the player rolls a total of 7 or 11, the player wins. If the player rolls a total of 2, 3, or 12, the player looses. For all other roll values, the player will repeatedly roll the pair of dice until either she rolls the initial value again (in which case she wins) or 7 (in which case she loses)  
  
Your class will support methods:

* newGame(): starts a new game by rolling a pair of dice. If the value (sum of two dice) of the roll obtained is 7 or 11, then a winning message is printed. If the value obtained is 2, 3 or 12, then a losing message is printed. For all other roll values, on the screen is printed a message telling the user to throw for point.
* forPoint(): generates a roll of a pair of dice and and depending on the value (sum of two dice) obtained prints one of three messages as appropriate (and shown below.)

Usage:  
>>> c = Craps()  
>>> c.newGame()  
Throw total: 3... you lost!  
>>> c.newGame()  
Throw total: 6. Throw for Point.  
>>> c.forPoint()  
Throw total: 8. Throw for Point.  
>>> c.forPoint()  
Throw total: 3. Throw for Point.  
>>> c.forPoint()  
Throw total: 6... you won!  
>>> c.newGame()  
Throw total: 6. Throw for Point.  
>>> c.newGame()  
Throw total: 6. Throw for Point.  
>>> c.newGame()  
Throw total: 2... you lost!

**4.**    Develop a class BankAccount that supports 4 methods:

* \_\_init\_\_(): initializes the bank account balance to the value of the input argument, or to 0 if no input argument is given.
* withdraw(): takes an amount as input and withdraws it from the balance.
* deposit(): takes an amount as input and adds it to the balance.
* balance(): returns the balance on the account.

Usage:  
>>> x = BankAccount(700)  
>>> x.balance())  
700.00  
>>> x.withdraw(70)  
>>> x.balance()  
630.00  
>>> x.deposit(7)  
>>> x.balance()  
637.00

**5.**    Read about the time module in the Python documentation or in section 4.5 of the textbook. Using this module, implement class Date that support methods:

* \_\_init\_\_(): constructor that takes no input but initializes the date oject to the date using the time module.
* display(): takes a format argument and displays the date in the requested format. The format argument is a string
  + ’MDY’ : MM/DD/YY (for example: 01/14/14)
  + ’MDYY’ : MM/DD/YYYY (for example: 01/14/2014)
  + ’DMY’ : DD/MM/YY (for example: 14/01/14)
  + ’DMYY’ : DD/MM/YYYY (for example: 14/01/2014)
  + ’MODY’ : Mon DD, YYYY (for example: Jan 14, 2014)

Hint: methods localtime() and strftime() from module time along with the type struct\_time might be useful.

Usage:  
>>> x = Date()  
>>> x.display('MDY')  
'01/14/14'  
>>> x.display('MODY')  
'Jan 14, 2014'