CMSC 143: Introduction to Object-Oriented Programming with Robots

Lab 2: Fruitful Functions

Due September 14, 2009

Submit a copy of your python program (cmsc143-lab2-LASTNAME.py) on moodle. Your program should have your name, email, and the date at the top of the file as a comment.

Learning Objectives

- Use functions that return values.
- Compose functions
- Write functions that return values.
- Write test cases for your functions.

Battery Capacity

Your robot uses 6 AA batteries. Each AA battery can range between 0–1.5V, so the robot can range between 0–9V. Write a function `batteryLeft()` that returns the battery capacity as a percentage between 0–100%. (FYI: `getBattery()` returns the battery voltage as a floating point between 0.0–9.0.)

Temperature

Write a function `userConvertTemp()` that asks the user (via `raw_input()`) for the temperature (in farenheit) and prints out the temperature in celsius, kelvin, rankine, and delisle. Also, write a function `tableTemps()` that prints a table of: absolute zero (-459.67°F), the freezing point of water (32°F) the boiling point of water (212°F) in celsius, kelvin, rankine, and delisle.

As helper functions, write six functions that convert between the different temperature scales. Use **only the equations below**, you should reuse your functions whenever possible. You might want to draw the function black-box diagrams (input/output) before you start programming.

```
celsiusToFarenheit(degreesInCelsius)     farenheitToCelsius(degreesinFarenheit)
farenheitToKelvin(degreesInKelvin)       kelvinToFarenheit(degreesInKelvin)
farenheitToRankine(degreesInFarenheit)   farenheitToDelisle(degreesInFarenheit)
```

	from Celsius	to Celsius
Farenheit	$F = C \times \frac{9}{5} + 32$	$C = (F - 32) \times \frac{5}{9}$
Kelvin	$K = C + 273.15$	$C = K - 273.15$

	from Kelvin	to Kelvin
Rankine	$R = K \times \frac{9}{5}$	$K = R \times \frac{5}{9}$

	from Delisle	to Delisle
Rankine	$R = 671.67 - D \times \frac{6}{5}$	$D = (R - 671.67) \times \frac{5}{6}$

Sanity Check

Write two functions: `testCelsius(degrees)` that takes a temperature and returns the result of the expression: `celsiustoFarenheit(farenheitToCelsius(x))` and another `testKelvin(degrees)` that takes a temperature and returns the result of the expression: `kelvinToFarenheit(farenheitToKelvin(x))`. What are the results for the following function calls?

```
testKelvin(-10)      testCelsius(2)
testKelvin(83.323)   testCelsius(43.5)
```

Finding Errors in an ISBN Error Detector

The ISBN (International Standard Book Number) is a universial identifier for a book. Invented in the 1966 it has since been adopted as an international standard. The ISBN has a built in error detection scheme. The last digit, the thirteenth digit, for the ISBN-13 version in 2005, is computed based on the first 12 digits. This digit can then be used to make sure the ISBN is valid.

We have been hired by amazon to make sure some of the ISBN's are correct. Below is a function for computing whether there is an error in the ISBN. But, it has an issue (or two), your task is to fix it! You should read the ISBN wikipedia page first. The bracket notation is grabbing characters from the string, `isbn[0]` grabs the first character `isbn[1]` grabs the second, etc.

```
def hasError():
    ''' purpose: asks the user to input an ISBN and computes what the error
                 digit is supposed to be.
        author: Keith O'Hara
        date: Sep 11 2009
    '''

    isbn = raw_input("Enter the ISBN of the book (without dashes): ")

    value = isbn[0]*1 + isbn[1] * 3
    value = value + isbn[2] * 1 + isbn[3] * 3
    value = value + isbn[4] * 1 + isbn[5] * 3
    value = value + isbn[7] * 1 + isbn[7] * 3
    value = value + isbn[8] * 1 + isbn[9] * 3
    value = value + isbn[10] * 1 + isbn[11] * 3

    remainder = value % 10

    error_code =  (10 - remainder) % 10

    print "The error code should be", error_code, "and you say its", isbn[12]

hasError()
```