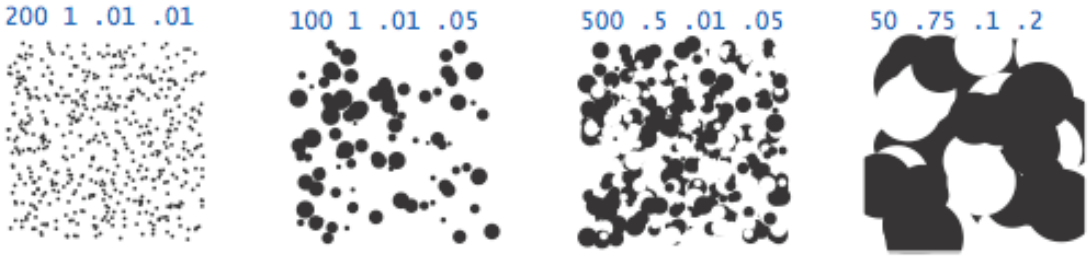
Faculty of Arts & Sciences
Department of Computer Science
CMPS 200—Introduction to Programming
Assignment 9 – Due November 17 (8:00AM), 2014

Notes

- You should write the appropriate static methods to show structure in your solution and to help you in writing other methods.
- Give meaningful names to methods and variables in your code.
- Include a comment at the beginning of your program with basic information and a description of the program and include a comment at the start of each method.

Problem 1 `Circles.java`

Write a program `Circles.java` that draws filled black and white circles of random size at random positions in the unit square, producing images like those below. Your program should take four command-line arguments: the number of circles, the probability that each circle is black, the minimum radius, and the maximum radius.
Note: the `DrawingPanel` could be of any size, as long as the graphic fits in it.



Problem 2 `LuckPercentage.java`

An integer greater than one is called a prime number if its only positive divisors (factors) are one and itself. For example, the divisors of 7 are 1 and 7; and the first six primes are 2, 3, 5, 7, 11 and 13.

In this problem, we introduce a prime character, which is a character with the corresponding ASCII code translating to a prime number. For example, the ASCII code of the character 'A' is 65, thus, 'A' is not a prime character. The ASCII code of the character 'C' is 67, therefore, 'C' is a prime character.

In this problem, your task is to define the ***percentage of luck*** for each name provided by the user. Names are constituted by characters with the ASCII code from 65 ('A') to 122 ('z'), including underscores.

We compute the percentage of lucky names by calculating the average weight of the characters constituting each name, where each character is given either the weight 100 if it is prime or the weight 0 otherwise. Given a name N of length L, the percentage of luck for N is computed as follow:

$$W(N) = \frac{\sum_{i=0}^{L-1} w_i}{L}$$

where, w_i is the weight of character at index i.

Write a program, `LuckPercentage.java`, which accepts from the command line a word and then calculates the percentage of luck associated with that name.

Example:

```
> java LuckPercentage SISCO
W("SISCO") = 100/5 + 100/5 + 100/5 + 100/5 + 100/5 = 100
Thus, the word SISCO is 100% Lucky
> java LuckPercentage cake
W("cake") = 0/4 + 100/4 + 100/4 + 100/4 = 75
Thus, the word cake is 75% Lucky
```

Problem 3 `Flip.java`

Write program that flips a coin until the same result occurs twice in a row. In other words, if a head is flipped followed by another head or if a tail is flipped followed by another tail, your method should end. Each time the coin is flipped, print *H* for heads or *T* for tails.

Example:

```
> java Flip
HTHH

> java Flip
THTHTT
```

Problem 4 `DigitRange.java`

Write a method named `digitRange` that accepts an integer as a parameter and returns the range of values of its digits. The range is defined as 1 more than the difference between the largest and smallest digit value. For example, the call of `digitRange(68437)` would return 6 because the largest digit value is 8 and the smallest is 3, so 8 - 3 + 1 = 6. If the number contains only one digit, return 1. *You should solve this problem without using strings*. The main method should obtain a number from the command line, invoke the `digitRange` method, and then report its result. *Example:*

```
> java DigitRange 68437

The digit range of "68437" is 6.
```

Problem 5 `Palindromes2.java`

Write a method `isPalindromePhrase` that accepts a string as argument and returns `true` or `false` indicating if the string is a palindrome or not. In this version of the method, whitespaces, commas, and apostrophes are ignored. For example, "Lonely Tylenol" and "Madam, I'm Adam" are both palindromes. The signature of the method should be:

```
public static boolean isPalindromePhrase(String str)
```

You are not allowed to generate a new string in your implementation of this method, and your iteration must be expressed with a **`while`** loop.

Use the method `isPalindromePhrase` to write a program `Palindromes2.java` that takes an integer *N* from the command line followed by *N* strings and prints the strings that are palindromes according to this extended definition.

Problem 6 `Discount.java`

Write a program `Discount.java` that reads in a list of prices from command line arguments and writes on the standard output a ***formatted*** table consisting of three columns as shown below. The first column consists of the prices as read from the command line (in US dollars). The second column prints a discounted price that represents a 1/3 discount on the price. The third column prints the equivalent value in Lebanese Liras rounded to the nearest LL 50 (assume a conversion rate $1 = LL 1508). Your output should have 2 digits after the decimal in the first 2 columns and no decimal in the third. It should also print column headers and makes sure the columns are lined up properly. An example invocation of the program might produce:

```
> java Discount 80.0 140.0
Price (in $USD)     Discounted   Equivalent Price (in LL)
$ 80.00             $53.33       LL  80450
$140.00             $93.33       LL 140750
```

Submission Instructions

- As usual, submit your commented source code and sample runs in a zip file named s#_asst9_*netid*, where # is your section number and *netid* stands for your AUBnet user name.