

# CSSSKL142 - Lab 5

## More Loops

University of Washington Bothell

### Summary

This lab contains four exercises. You must complete all exercises within the allotted time for the lab. Please write clear and well commented code. Topics covered today: More Loops and (File I/O next week). We will also continue to practice and improve on branching and keyboard input with Scanner.

In a class called Lab5, implement all of the following methods as specified.

### Part 1

Suppose you find a magic \$1.00 coin. Its magic power is as follows: as each day passes, you get an additional dollar plus half of what you already had (it appears by the window somehow). Write a method called **getRichQuick** (no input is necessary for this method) that prints the first  $n$  days while your total is less than \$1,000,000 dollars. In other words, how many days does it take for you to earn \$1,000,000? Your program should calculate these numbers and print the following output:

```
Day 1: $1
Day 2: $1 + ($1 + .50) = $2.50
Day 3: $2.50 + ($1 + 1.25) = $4.75
...
Day N: $X + ($1 + Y) >= $1000000
```

### Part 2

Write a method called **eTaylor** that takes as input (i.e. argument) a **double x** and returns **the value of  $e^x$** . This method should NOT use any Math library functions, so you will need to use Taylor Series for  $e^x$  (loop until terms become close to 0, say  $10e-16$ ):

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

*Hint:* Write a helper method for finding the  **$n^{\text{th}}$  power of a number  $x$** , divided by  $n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1$  that is, your helper method should accept an  $x$  and output  $\frac{x^n}{n!}$ . Your method **eTaylor** will return the value of

$$\sum_{n=0}^{\infty} \frac{x^n}{n!}$$

*Hint:* Try the series at least to the point where  $n = 5$ ,  $n = 10$  and  $n = 20$ .

### Part 3

**Hint: User will enter certain number of random words, you need to read the input as string and evaluate based on the following instructions.**

Write a void method called **palindromeCheck** that takes NO argument. The method should have functionality to check whether or not the word is a palindrome and print to the screen all of the palindromes, one per line.

Also, the last line of the output should have the message: “**There are x palindromes out of y words provided by user**” (where x is the number of palindrome words detected, and y is the total number of words entered by user). Hint: for this lab exercise you will need the following methods on String objects: **length()** gives the length of a string (that is, the number of characters it contains) and **charAt(i)** - gives the character at position i. For example:

```
int sizeOfString = "bob".length()           //should be 3
char firstChar = "bob".charAt(0)           //should be 'b'
```

## Part 4

In main, write a menu that allows a user of your program to select and run the programs you wrote in the methods above. The user should be able to play with your methods a many times as they want and only exit the program if they want to exit. For example:

```
Welcome to Lab5!
Enter 1 to check how long it takes to get rich on a magic dollar coin.
Enter 2 to calculate e^x for any real x.
Enter 3 to enter palindrome words.
Enter 4 to re-print the menu.
Enter 0 to exit.
What is your choice?
3
Enter an x:
1
e^1 = 2.7182818284590452

What is your choice?
0
Thanks for participating! Goodbye.
```

Note that the methods from part 2 and 3 did not have you asking for keyboard input. Do NOT go back and modify your methods from part 2 and 3, instead handle the request for keyboard input (to get the x or all the words) in main after the user has made his/her choice (i.e. item 2 or 3 in the menu).

## Final Part

Remember to submit your **Lab5.java** code to Canvas! Also, don't forget to comment your code and practice good spacing!