

CPSC 1150 – Program Design

Assignment #3

Total Marks: 52 marks (2 marks are bonus marks)

Goals

This assignment gives students more experience on:

- Problem Solving (especially with Strings, loops, and functions)
- Writing Algorithms, using flowchart or pseudocode
- Testing Algorithms
- Writing Java programs

As a rule of thumb any function in java including main function should be around 20 lines or less. If it is more, you should split your function in two or more functions.

Problem Description

Exercise 1: [10 marks] Write a java program named **Count.java** that prompts the user to enter a string and displays the number of the uppercase letters and the number of digits in the string. Here is a sample run:

```
Enter a string: Welcome to Java 8 (1.8)
The number of uppercase letters is 2
The number of digits is 3
```

Exercise 2: [12 marks] Write a java program named **SSN.java** that prompts the user to enter a Social Security Number in format of DDD-DDD-DDD, where D is a digit. The first digit cannot be zero. Make sure that second set of three digits is more than 100. Your program should check whether the input is valid. Here are sample runs:

```
Enter a SSN: 123-268-097
123-268-097 is a valid social security number
```

```
Enter a SSN: 023-289-097
023-289-097 is an invalid social security number
```

```
Enter a SSN: 198-068-097
198-068-097 is an invalid social security number
```

```
Enter a SSN: 198-1680-97
198-1680-97 is an invalid social security number
```

Exercise 3: [15 marks] A palindromic prime is a prime number whose reversal is also a prime. For example, 131 is a prime and a palindromic prime, as are 757 and 353. Write a program named **PalindromPrime.java** that displays the first 100 palindromic prime numbers. Display 10 numbers per line in a tabular format as follows (left justified):

2	3	5	7	11	101	131	151	181	191
313	353	373	383	727	757	787	797	919	929
...									

Use functions such as isPrime(x), isPalindrom(x), etc. to have a stylish program.

Exercise 4: [15 marks] In lab4, you wrote a program that implements one round of rock, scissor, and paper game. Revise this game for more rounds (at least 5 rounds or more) to let the user continuously plays until either the user or the computer wins more than two rounds than its opponent. (RSPGame.java)

Use functions to implement your algorithms.

Here is a sample run:

```
Round 1: Select rock (0), scissor (1), or paper (2): 0
The computer is rock. You are rock too. It is a draw.
```

```
Round 2: Select rock (0), scissor (1), or paper (2): 1
The computer is paper. You are scissor. You won!
```

```
Round 3: Select rock (0), scissor (1), or paper (2): 0
The computer is paper. You are rock. Computer won!
```

```
Round 4: Select rock (0), scissor (1), or paper (2): 2
The computer is scissor. You are paper. Computer won!
```

```
Round 5: Select rock (0), scissor (1), or paper (2): 1
The computer is rock. You are scissor. Computer won!
```

```
Game Over! Computer is the winner! Try again!
```

Grading

The programming questions are marked based on correctness, style of program and documentations. Add the external documentations in **answers.pdf**.

Submission

Submit a zip file named **Firstname-Surname-StudentId.zip** including **answers.pdf**, **Count.java**, **SSN.java**, **PalindromPrime.java** and **RSPGame.java** on Brightspace.