CSE3040: Java Programming (Fall 2020)

Homework 1

Due: October 6, 11:59PM (KST)

※ Write a Java code for each problem. Make sure your program satisfies all specified requirements. Carefully read the submission instructions at the end of the document.

Problem 1 (20 points)

- Description

- Write a Java program that asks user for an alphabet and prints the ASCII code of the alphabet.

- The valid user input is a **single uppercase letter (A-Z), or a single lowercase letter (a-z)**.

- If the input is multiple characters, characters other than alphabets, or nothing, your program should output an error message and terminate.

- Your program should produce results as shown in the example output.

- Example output

```
ASCII code teller. Enter a letter: C
The ASCII code of C is 67.
```

```
ASCII code teller. Enter a letter: hello
You must input a single uppercase or lowercase alphabet.
```

Problem 2 (20 points)

- Write a Java program that plays a number guessing game with the user.

- When the program begins, the program draws a random number from 1 to 100.

- Then, the program asks the user to guess a number between 1 and 100.

- If the user input is smaller than the correct answer, print "Too small!" and ask again.

- If the user input is larger than the correct answer, print "Too large!" and ask again.

- If the user input is correct, print "Correct! Number of guesses: 3" and terminate the program. In order to do this, you need to count the number of guesses.

- Whenever the user gives a wrong answer, the program updates the range where the user is supposed to guess a number.

- If the user inputs a number that is not in the given range, print "Not in range!" and ask again. This case is not included in the number of guesses.

- For this problem, you can safely assume that the user will always enter a number. The user will not enter other things like alphabets, and the user will not give an empty input.

- Your program should produce results as shown in the example output.

- Example output

```
[1] Guess a number (1-100): 37
Too small!
[2] Guess a number (38-100): 53
Too large!
[3] Guess a number (38-52): 44
Correct! Number of guesses: 3
```

```
[1] Guess a number (1-100): 58
Too small!
[2] Guess a number (59-100): 35
Not in range!
[2] Guess a number (59-100): 85
Too small!
[3] Guess a number (86-100): 100
Too large!
[4] Guess a number (86-99): 98
Correct! Number of guesses: 4
```

Problem 3 (20 points)

- Write a Java program that counts instances of a letter in a text.

- When the program begins, the program asks user for a text.

- Once the user inputs the text, then the program asks user for a letter.

- The program prints how many instances of the given letter are there in the text.

- For the second prompt which asks user to enter a letter, if the user enters nothing or more than one characters, the program should print an error message and show the prompt again.

- For the letter counting, you should distinguish lowercase and uppercase letters. 'A' is different from 'a'.

- Your program should produce results as shown in the example output.

- Example output

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a letter: n
There are 5 n's in the text.
```

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a letter: z
There are 0 z's in the text.
```

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a letter: Hello
You must enter a single letter.
Enter a letter: H
There are 1 H's in the text.
```

Problem 4 (20 points)

- Write a Java program that counts instances of a string in a text.

- When the program starts, the program asks user for a text.

- Once the user inputs the text, then the program asks user for a string.

- The program prints how many instances of a given string are there in the text.

- For this problem, if the user inputs nothing on the second prompt, your program should print an error message and show the prompt again.

- Your program should produce results as shown in the example output.

- Example output

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a string: am
There are 2 instances of "am".
```

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a string: that
There are 0 instances of "that".
```

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a string:
You must enter a string.
Enter a string: m
There are 3 instances of "m".
```

Problem 5 (20 points)

- Write a Java program that finds students with the best exam scores.

- When the program starts, the program asks users for exam scores of five students.

- Then, the program prints the 1st and 2nd place students and their scores.

- You can safely assume that there are no two students with the same score. Also, the user will always enter a single integer.

- Your program should produce results as shown in the example output.

- Example output

```
Enter exam scores of each student.
Score of student 1: 50
Score of student 2: 70
Score of student 3: 30
Score of student 4: 90
Score of student 5: 40
The 1st place is student 4 with 90 points.
The 2nd place is student 2 with 70 points.
```

※ Java Naming Convention

When you write Java code it is good to follow the Java Naming Convention, which is a rule for naming variables, constants, methods, and classes.

(1) Variable Names

Start with a lowercase letter and use uppercase letters as separates. Do not use under bars (‘_’).

```
int myVar;
```

(2) Constant Names

Use all capital letters and use under bars as separators.

```
final int MY_CONST = 1;
```

(3) Method Names

Start with a lowercase letter and use uppercase letters as separators. Do not use under bars.

```
int myMethod()
```

(4) Class Names

Start with an uppercase letter and use uppercase letters as separators. Do not use under bars.

```
Public class MyClassName
```

※ Submission

※ Carefully read this part and follow the instructions. Not properly following the instructions here may result in point deduction.

(1) For this homework, you are going to submit only the .java files. You are going to submit one .java file for each problem.

(2) For problem 1, your file name should be Problem1.java. This means that your public class name should also be Problem1. For other problems, you should name your .java files this way.

(3) Once you are ready to submit, you should make the .java files into a single zip file named cse3040_hw1_20180001.zip. The numbers in the file name should be your student ID.

When you make the zip file, make sure that you are not using subfolders inside the zip file. When the zip file is extracted, the .java files should appear without having any subdirectory structures.

(4) Submit your zip file on the cyber campus.

※ Evaluation Criteria

Your solution to each problem will be tested with various test cases. You will get full points if your program passes all tests. If not, points will be given based on percentage of test cases passed.

For this homework, each problem is worth 20 points. The perfect score is 100.

※ Academic Integrity

- You should write your own code. You can discuss ideas with other students but must not copy their work. You can also get help from the Internet, but you must not copy the source code from the Internet either. We have a duplicate check program which tests whether your source code is similar to other students' code as well as codes that are on the Internet.

- Duplicate work will receive zero grade.

※ Late Policy

- 10% of the score is deducted for each day, up to three days. Submissions are accepted up to three days after the deadline.