UNIVERSITI MALAYSIA SARAWAK

Faculty of Computer Science and Information Technology

Student Name	Student ID Number	Signature
Song Wang Ye	71680	Swye
Ho Wan Yu	69860	Wan yu
Christopher Sii How Chiong	69385	Chiong

Subject Code: **TMF2954**	Subject Name: **Java Programming**
Assignment Title: **Group Assignment**	Lecturer: **Dr Amelia Jati Anak Robert Jupit**
Due Date: **23/5/2021**	Date Submitted: **23/5/2021**

Plagiarism and Collusion are methods of cheating that falls under Peraturan Akademik Universiti Malaysia Sarawak para 11: Etika Akademik

Plagiarism

Plagiarism is the presentation of work which has been copied in whole or in part from another person's work, or from any other source such as the internet, published books or periodicals without due acknowledgement given in the text.

Collusion

Collusion is the presentation of work that is the result in whole or in part of unauthorized collaboration with another person or persons.

Where there are reasonable grounds for believing that cheating has occurred, the only action that may be taken when plagiarism or collusion is detected is for the staff member not to mark the item of work and to report or refer the matter to the Dean. This may result in work being disallowed and given a fail grade or if the circumstances warrant, the matter may be referred to a Committee of inquiry for investigation. Such investigation may result in the matter being referred to the University Discipline Committee, **which** has the power to exclude a student.

Upon placing signature above, I certify that I have not plagiarized the work of others or participated in unauthorized collusion when preparing this assignment.

I also certify that I have taken proper case in safeguarding my work and have made all reasonable efforts to ensure that may work not be able to be copied.

MARK :

TMF 2954 Java Programming / TMF 2634 Java for Multimedia Programming
Group Assignment [15%]
DUE DATE: 23 May 2021, 11.55PM

Instruction:
Write a Java program that contain the following class, i.e., class **SecretCode**. Please refer to the given UML diagram for the minimum required class's data fields and methods. You are free to add additional class members as you see fit.

class **SecretCode**
-**inString**: String -**stringLen**: int -**outString**: String
+**SecretCode**() +**SecretCode**(**inString**: String) +**getStringLen**(**inString**: String): int +**checkStringValidity**(**inString**: String): boolean +**shiftChar**(**inString**: String, **stringLen**: int): String

The program prompts for a String literal (of any random length) and echoes an encoded String literal.

The method **shiftChar()** handles the encoding process by executing the following steps:

Step 1: Accept a string literal, i.e., **inString**.
Step 2: Perform a check (using method **checkStringValidity()**) to ensure the String literal ONLY contain consonants, numbers 1-5 or whitespace, if this condition is not met, prompt user for a new input.
Step 3: Compute the number of characters inside **inString**, excluding whitespace(s), i.e., stringLen.
Step 4: For each character inside **inString**, shift it forward *n* position(s), where *n* equals to the number of characters (excluding whitespace(s)) inside **inString**.

index	0	1	2	3	4	5	6	7	8	9
char	1	b	c	d	2	f	g	h	3	j

index	10	11	12	13	14	15	16	17	18	19
char	k	l	m	n	4	p	q	r	s	t

index	20	21	22	23	24	25	26
char	5	v	w	x	y	z	ws

Table 1

The string literal DOES NOT ACCEPT vowels. Instead it will only accept '1' for 'a', '2' for 'e', and so forth as shown in Table 1 above.

For example, instead of 'zoo kch', we have a String literal “z44 kch” that has a length of 6 characters (minus whitespace), i.e., **stringLen** = 6. Thus, the encoded output is:

z→ws→1→b→c→d→2
4→p→q→r→s→t→5
4→p→q→r→s→t→5
ws→1→b→c→d→2→f
k→l→m→n→4→p→q
c→d→2→f→g→h→3
h→i→j→k→l→m→n

The encoded String literal: “255fq3n”

Step 5: The method returns the encoded String literal, i.e., **outString**.

Another example of a String literal for "dont cheat" is "d4nt ch21t" which gives a **stringLen** = 9 with the encoded output of "mxwb3lqnjb".

Sample Output:

```
Enter string:
z44 kch
inString: z44 kch
len: 6
outString: 255fq3n
```

```
Enter string:
d4nt ch21t
inString: d4nt ch21t
len: 9
outString: mxwb3lqnjb
```

Submission instruction:

1. Name your main class file as Assignment.java. This is the file that I will look for to compile and execute. If you don't have this file, you will get 0.
2. You would also need to hand in the faculty assignment cover page (**.pdf only**) with the signature of each member agreeing to the terms written on the cover page.
3. **Zip** (not .rar) your .java file(s) and the cover page with the following naming convention, G0X_Groupname.zip. For example, G01_MOTS.zip.
4. Submit your .zip file through the link on the assignment submission page.
5. Fill in the peer evaluation form in Week 9 (will be uploaded before the due date) according to your group number.
6. Late submission will result in a penalty. If your group member did not contribute to the assignment at all, he/she will get 0 marks for the entire assignment.

Marking rubric (15%)

		Level A	Level B	Level C	Level D
		0 mark	1 mark	3 marks	5 marks
CLO1 (5%)	Output	No output or the program does not compile/run at all.	The program echoes the inputted String literal.	Level B and echoes the String literal's length.	Level C and echoes the correct output.
CLO2 (5%)	Class declaration & Algorithm	Missing at least one of the minimum class members.	Minimum class members are all implemented.	Level B and checkStringValidity() is working.	Level B and shiftChar() is working.
CLO3 (5%)	Ability to work in a team	Less than 4 members in a group.	2-3 members in a group.	Level B and submission met the requirements: on-time, follows submission format, faculty assignment cover etc.	Level C and received at least 4 marks for peer grading.

THE END