Concordia University
Computer Science & Software Engineering
Faculty of Engineering and Computer Science

Concordia University
COMP 248 – Summer 2019
Assignment 2

Due Date:	By 11:55pm Wednesday June 5, 2019
Evaluation:	6% of final mark (see marking rubric at the end of handout)
Late Submission:	none accepted
Purpose:	The purpose of this assignment is to help you learn Java identifiers, assignments, input/output nested if and if/else statements, switch statements and non-nested loops.
CEAB/CIPS Attributes:	Design/Problem analysis/Communication Skills

General Guidelines When Writing Programs:
Refer to handout of assignment 1 for details.

Question 1 – Uno deck of cards & switch statement (7 pts)

"**Uno** (/ˈuːnoʊ/; from Italian and Spanish for 'one'; stylized as **UNO**) is an American shedding-type card game that is played with a specially printed deck. The game's general principles put it into the Crazy Eights family of card games. The game was originally developed in 1971 by Merle Robbins in Reading, Ohio, a suburb of Cincinnati." (https://en.wikipedia.org/wiki/Uno_(card_game))

The deck has 108 cards and is a combination of the following cards:

1. There are 4 colors: red, blue, green and yellow numbered from 0 to 9

http://rulesoftheinternet.com/uno-rules.html

2. There are the following action cards

https://www.unorules.com/

Notice that the 2 wild cards do not have a colour.

Write a Java program using *switch* statements whenever possible instead of nested if statements, that takes user input describing a playing card (value and color) in the following shorthand notation into a single String variable:

	Notation	Meaning
Value	0 …. 9	Card values
	W4	Wild Draw 4
	W	Wild
	D	Draw 2
	S	Skip
	V	Reverse
Colour	R	Red
	B	Blue
	G	Green
	Y	Yellow

Note: The *Wild Draw 4* card is represented by *W4* with no color and the *Wild* card is represented as *W* with no color.

The input must be 2 character long, unless it is the simple *Wild* card in which case it is one-character long.

Your program should then display the full description of the card. Your program should indicate if the requested card is not a valid Uno card, for example if it is 1-character long and not a *W*, or is 2-characters long but not one of the cards found in the deck (Figure 2) or it is not entered in the correct format such as the length is greater than 3 (Figure 4). Your program should accept lower and upper case input.

Following are some sample outputs (Figures 1 to 5) to illustrate the expected behaviour of your program. Your program should work for any input, not just the ones in the samples below. The text in **green** is user input. You can change the format and/or content of the output as long as it still conveys the same information.

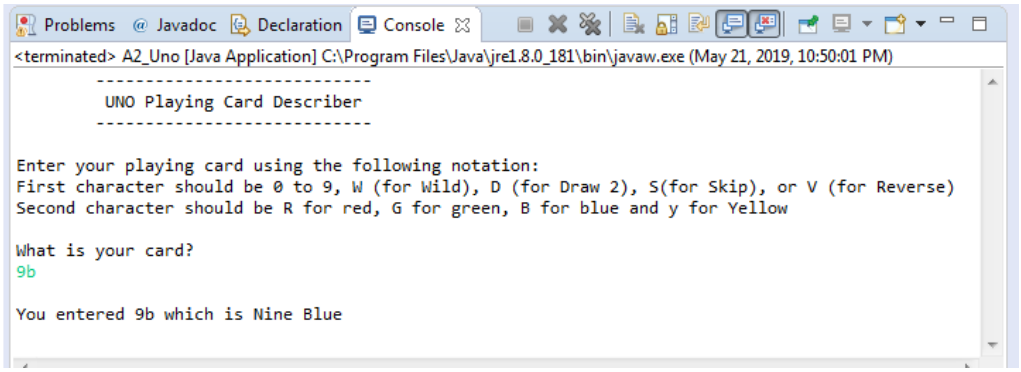


Figure 1- Valid card

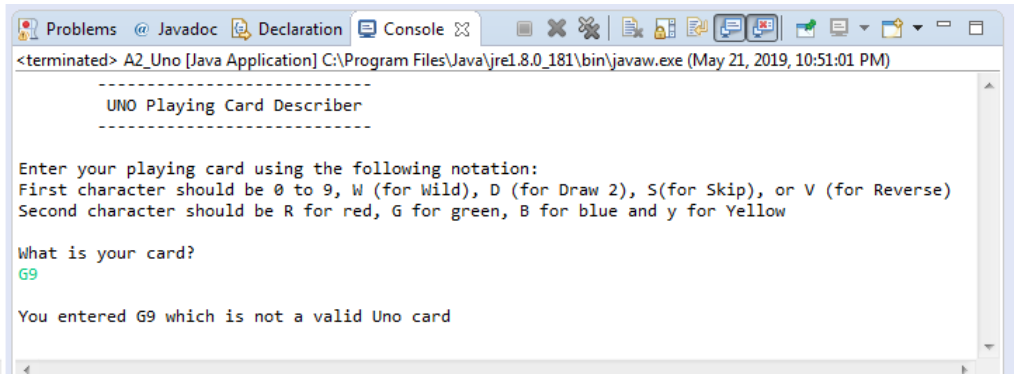


Figure 2- Invalid card

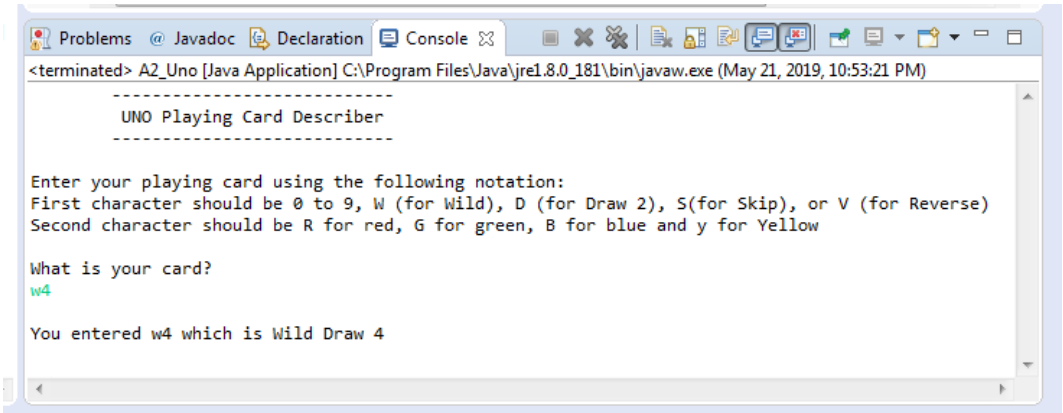


Figure 3- Wild Draw 4 card

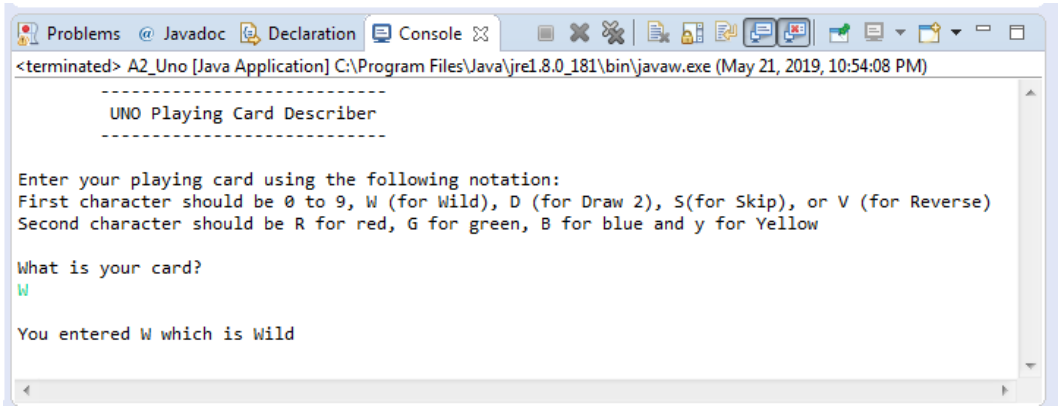


Figure 4- Simple Wild card

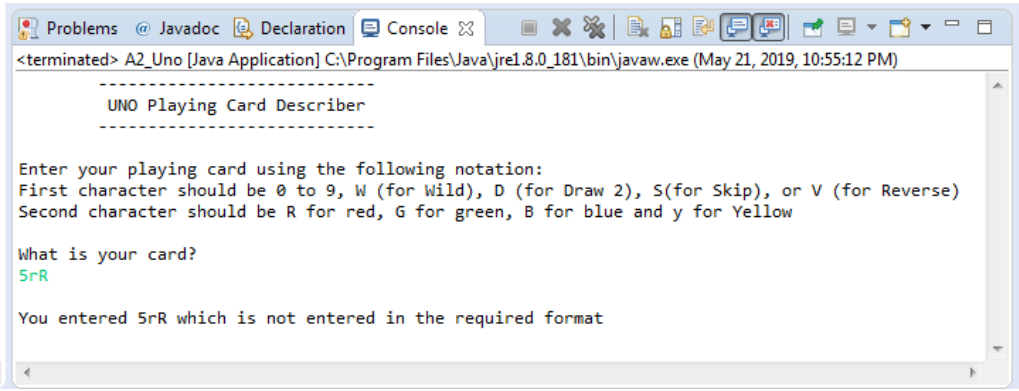


Figure 5- Incorrect format - Too many characters

Question 2 – String variables/Selection & loops. (8.5 points)

Write a complete Java program which prompts the user for a sentence on one line where each word is separated by one space, reads the line into one String variable using *nextline()*, converts the string into Ubbi dubbi and displays the translated sentence.

Ubbi dubbi works by adding *ub* before each vowel sound in a syllable. For example, *hello* becomes *h**ub**ell**ub**o*. The word *speak* has the vowel sound ea, so in Ubbi dubbi it becomes *sp**ub**eak*.

Tidbit: For those who are fans of The Big Bang Theory, Ubbi dubbi was used between Penny and Amy in season 10 episode 7 as a means of having a secret conversation, to counter Sheldon and Leonard's Klingon. Link to the video:
https://www.youtube.com/watch?v=rfR03gibh6M&feature=youtu.be&t=54s)
(From https://en.wikipedia.org/wiki/Ubbi_dubbi)

Modified Ubbi dubbi rules for this assignment

To simplify things for this assignment,

1. The vowels are a, e, i, o and u.
2. In the case that a word has 2 vowels following one another we only place *ub* in front of the 1st vowel. So zoom would be *z**ub**oom* and not *z**ub**o**ub**om*. Similarly, *steak* would be *st**ub**eak* and not *st**ub**e**ub**ak*. (Figure 9)
3. For words that are 1 character or 2 character long we add ub in front of each character whether they are vowels or consonants. (Figure 7)
4. If *e* is the last character of the word, we don't add *ub* in front of it. For example, the word *one* is translated as ***ub**one* and not ***ub**on**ub**e*.(Figure 9, last word)
5. Any other characters (punctuations, digits) are treated as non-vowels. (Figure 8)

Assumptions.

For this assignment you can assume that

1. There is exactly 1 space between words.
2. The entered string does not start nor end with a space.
3. All words are in lower case letters.
4. The sentence has at least one non-blank character.

Following are some sample outputs (Figures 6 to 9) to illustrate the expected behaviour of your program. Your program should work for any input, not just the ones in the samples below. The text in **green** is user input. You can change the format and/or content of the output as long as it still conveys the same information.

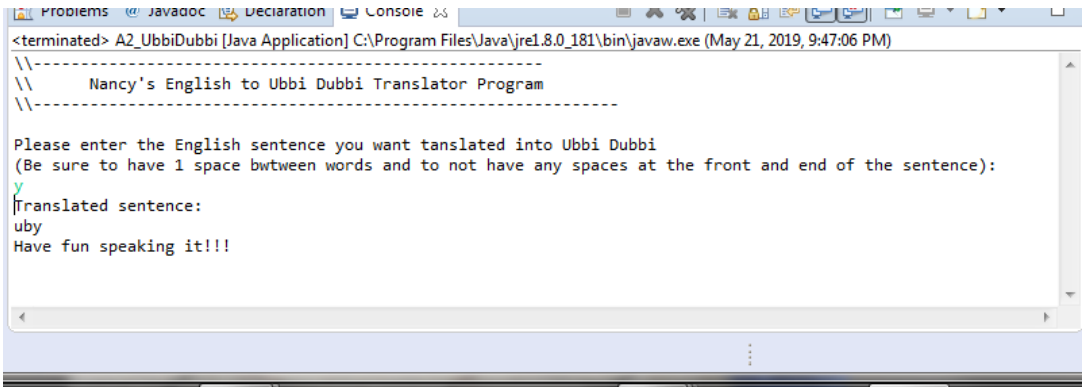


Figure 6 - Example with a single 1-character word sentence

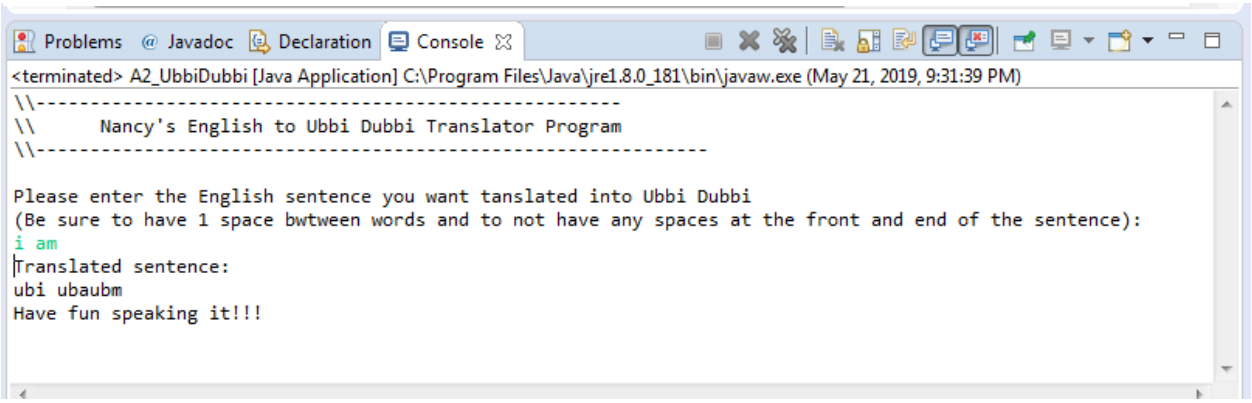


Figure 7 - Example with 1-character and 2-characters long words

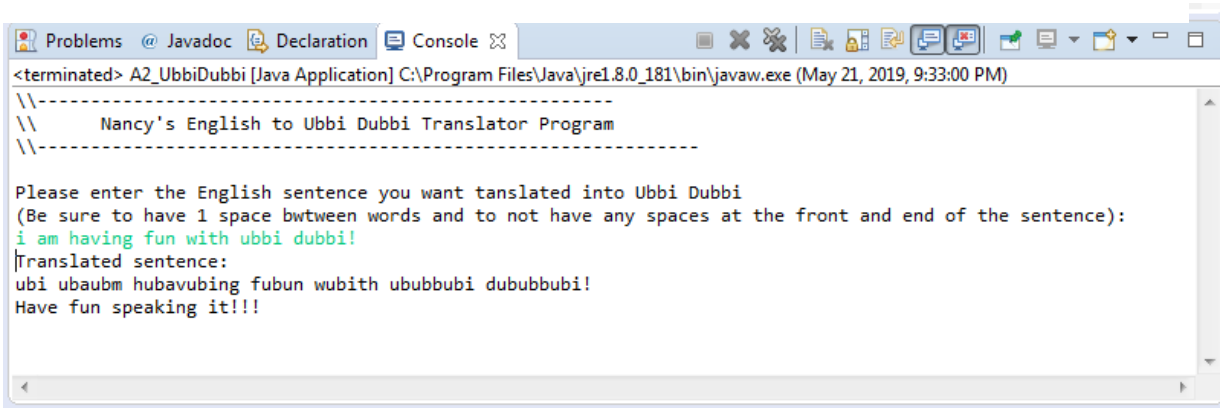
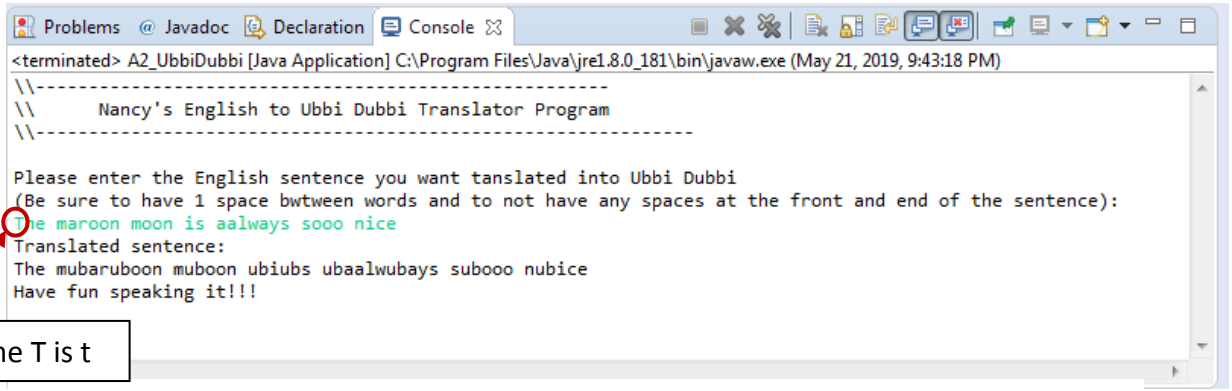


Figure 8- Example with mixed words and punctuation



Assume T is t

Figure 9- Example with words with 2 or more vowels following each other and a word that ends in e

Submitting Assignment 2

1. Compress the source codes (the .java file only please) of this assignment together into one file following the following naming convention:

   The zip file should be called *a#_studentID*, where # is the number of the assignment and *studentID* is your student ID number. For example, for this assignment, student 123456 would submit a zip file named a2_123456.zip.
   You will have 2 .java files to submit for this assignment

2. **Upload** your compressed file using the appropriate assignment link in the Assessment page.

3. **NOTE: The only compressed file format we accept is .ZIP. No .RAR files are accepted.**

Evaluation Criteria for Assignment 1 (20 points)

Source Code	
Comments for all 3 questions (1.5 pts.)	
Description of the program (authors, date, purpose)	0.5 pts.
Description of variables and constants	0.5 pts.
Description of the algorithm	0.5 pts.
Programming Style for all 3 questions (3 pts.)	
Use of significant names for identifiers	1 pt.
Indentation and readability	1 pt.
Welcome Banner or message/Closing message	1 pt.
Question 1 (7 pts.)	
Use of switch where possible	2 pts.
Recognizing invalid card	1 pt.
Recognizing incorrect format	1 pt.
Accepting upper/lower case input	1 pt.
Displaying correct description	2 pts.
Question 2 (8.5 pts.)	
ub in front of a single vowel	1 pts.
Only one ub when more than 1 vowel	2 pts.
Single character and 2-character words	2 pts.
e last letter of a word	2 pts.
Non-vowel characters	1.5 pts.
TOTAL	**20 pts.**