Lab Session 13

**As usual, if you have not finished a previous lab, make sure to complete those before getting to these questions.**

**Part 1: Theory**

1. Given these declarations and initializations:

char c = ‘d’;

char d = ‘3’;

To what will these expressions evaluate?

c >= ‘a’ && c <= ‘z’

c + 4

d + 2

1. Assume you have a char variable *c*. Write a Boolean expression to check if the character stored in *c* represents a letter.
2. (Bonus: Try this on your own) Write an algorithm to print the first 1000, negative numbers (i.e. -1,-2,-3,-4,-5,…) We will be looking at repetition in the next class. (Hint: How do you keep track of when to stop the program?)

Part 2: Application

1. Write a method in Java called *isSquare*. This method must take values of length and breadth of a rectangle as input, check if it is square or not and return a boolean. In the main method, get the values of length and breadth from the user and print if it is a square or not. (Note: use integer for length and breadth and not double)
2. A school has following rules for grading system:

a. Below 25 - F

b. 25 to 45 - E

c. 45 to 50 - D

d. 50 to 60 - C

e. 60 to 80 - B

f. Above 80 - A

Write a method called *getGrade* that takes as input a double value representing the marks and return a character representing the corresponding grade. In the main method, ask the user to enter marks and print the corresponding grade. If the marks entered is not in any of the above listed range, your program must print "Invalid marks!” and end. (Note: All the print statements must go in the main method and not in the getGrade method).

1. Write a method *encrypt* that takes as input a character and “encrypts” it by shifting the alphabet by 3. for example,

a becomes d

b becomes e

c becomes f etc

The method should return the encrypted character. In the main method, get a character from the user as input and print the encrypted character.

Hint: There is no method with scanner called nextCharacter. However, you can use the next() method to get a String. Any time you have a String, you can get the first character of the String via a method *.charAt(0).*

Example: String word = reader.next(); // assumes reader was declared already

char firstLetter = word.charAt(0);

Note that other subsequent characters can be picked out by changing the 0 to another number. For example, word.charAt(1) would get the second character, word.charAt(2) would get the 3rd, etc.

**Character conversion**

1. Write a method *toLowerCase* that takes as input a character and returns a character that is the uppercase version. If the character is not a valid letter, then the method should just return the same character.

In the main method, ask the user to enter a single letter. Print the character returned in the main method

For example, if the user types in the character *C*, your program you print c. If the user types in *d* then the program should print d. If the user types in [, your program should print [.