EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
CSE110: Object Oriented Programming

Assignment	**: 02**
Instructor	**: Ahmed Abdal Shafi Rasel, Lecturer, Department of CSE**
Section	**: 15, 16**
Trimester	**: Spring 2024**

Objective: The objective of this assignment is to develop problem solving skills relating to creating methods, method-overloading, class, object, constructor, constructor overloading, access modifiers, instance and static methods, immutable object, etc.

Tasks:

No.	Problems

1. (Palindrome integer) Write the methods with the following headers:

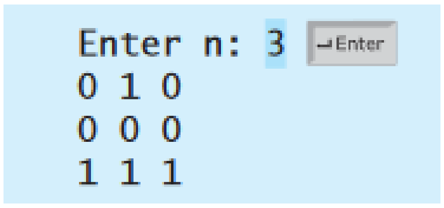
```
public static int reverse(int number)
public static boolean isPalindrome(int number)
```

Use the reverse method to implement isPalindrome. A number is a palindrome if its reversal is the same as itself. Write a test program that prompts the user to enter an integer and reports whether the integer is a palindrome.

2. (Display matrix of 0s and 1s) Write a method that displays an n-by-n matrix using the following header:

```
public static void printMatrix(int n)
```

Each element is 0 or 1, which is generated randomly. Write a test program that prompts the user to enter n and displays an n-by-n matrix. Here is a sample run:



```
Enter n: 3 ↵Enter
0 1 0
0 0 0
1 1 1
```

3. (Check password) Some websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:

- ■ A password must have at least eight characters.
- ■ A password consists of only letters and digits.
- ■ A password must contain at least two digits.

Write a program that prompts the user to enter a password and displays Valid Password if the rules are followed or Invalid Password otherwise.

4. (Count the letters in a string) Write a method that counts the number of letters in a string using the following header:

```
public static int countLetters(String s)
```

Write a test program that prompts the user to enter a string and displays the number of letters in the string.

5. (Occurrences of a specified character) Write a method that finds the number of occurrences of a specified character in a string using the following header:

```
public static int count(String str, char a)
```

For example, count("Welcome", 'e') returns 2. Write a test program that prompts the user to enter a string followed by a character and displays the number of occurrences of the character in the string.

6. Design a class named Stock that contains:

- ■ A string data field named **symbol** for the stock's symbol.
- ■ A string data field named **name** for the stock's name.
- ■ A double data field named **previousClosingPrice** that stores the stock price for the previous day.
- ■ A double data field named **currentPrice** that stores the stock price for the current time.
- ■ A constructor that creates a stock with the specified symbol and name.
- ■ A method named **getChangePercent()** that returns the percentage changed from **previousClosingPrice** to **currentPrice**.

Draw the UML diagram for the class and then implement the class. Write a test program that creates a Stock object with the stock symbol ORCL, the name Oracle Corporation, and the previous closing price of 34.5. Set a new current price to 34.35 and display the price-change percentage.

7. (Use the GregorianCalendar class) Java API has the **GregorianCalendar** class in the java.util package, which you can use to obtain the year, month, and day of a date. The no-arg constructor constructs an instance for the current date, and the methods ***get(GregorianCalendar.YEAR)***, ***get(GregorianCalendar.MONTH)***, and ***get(GregorianCalendar.DAY_OF_MONTH)*** return the year, month, and day.

Write a program to perform two tasks:

■ Display the current year, month, and day.
■ The GregorianCalendar class has the **setTimeInMillis(long)**, which can be used to set a specified elapsed time since January 1, 1970. Set the value to 1234567898765L and display the year, month, and day.

8. (Stopwatch) Design a class named StopWatch. The class contains:

■ Private data fields startTime and endTime with getter methods.
■ A no-arg constructor that initializes startTime with the current time.
■ A method named start() that resets the startTime to the current time.
■ A method named stop() that sets the endTime to the current time.
■ A method named getElapsedTime() that returns the elapsed time for the stopwatch in milliseconds.

Draw the UML diagram for the class and then implement the class. Write a test program that measures the execution time of sorting 100,000 numbers using selection sort.

9. (Algebra: 2 * 2 linear equations) Design a class named LinearEquation for a 2 * 2 system of linear equations:

$$\begin{aligned} ax + by &= e \\ cx + dy &= f \end{aligned} \qquad x = \frac{ed - bf}{ad - bc} \qquad y = \frac{af - ec}{ad - bc}$$

The class contains:

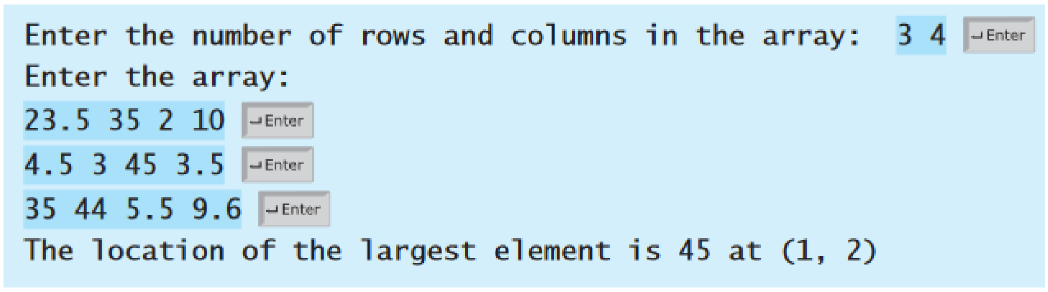
■ Private data fields a, b, c, d, e, and f.
■ A constructor with the arguments for a, b, c, d, e, and f.
■ Six getter methods for a, b, c, d, e, and f.
■ A method named **isSolvable()** that returns true if ad - bc is not 0.
■ Methods **getX()** and **getY()** that return the solution for the equation.

Draw the UML diagram for the class and then implement the class. Write a test program that prompts the user to enter a, b, c, d, e, and f and displays the result. If ad - bc is 0, report that “The equation has no solution.”.

10. (The Location class) Design a class named Location for locating a maximal value and its location in a two-dimensional array. The class contains public data fields **row**, **column**, and **maxValue** that store the maximal value and its indices in a two dimensional array with row and column as int types and **maxValue** as a double type.

Write the following method that returns the location of the largest element in a two dimensional array: **public static Location locateLargest(double[][] a).**

The return value is an instance of Location. Write a test program that prompts the user to enter a two-dimensional array and displays the location of the largest element in the array. Here is a sample run:



```
Enter the number of rows and columns in the array:  3 4
Enter the array:
23.5 35 2 10
4.5 3 45 3.5
35 44 5.5 9.6
The location of the largest element is 45 at (1, 2)
```