PROGRAMMING LAB

ASSIGNMENT NO. 1

Date: January 19, 2017

Submission Date: **February 10, 2017**

Follow the below Instructions to do the programs. **Violation of these instructions will causes penalization.**

- ❖ Exhibit your programs in complete object-oriented format only.
- ❖ Your programs should ensure that at least some of the OOP principles must exists in the code like Class, Object, Data Encapsulation, and Data Abstraction. You may implement some programs with Polymorphism and Inheritance if it requires.
- ❖ Suggesting you to use End-of-line style format for coding (or you may follow New-line style format).
- ❖ Provide appropriate documentation in the coding for all programs (Explanation for the program in short description format).
- ❖ Naming to the variables and methods also in proper manner (instead of giving short names).

1 (*Population projection*) The U.S. Census Bureau projects population based on the following assumptions:

- ■ One birth every 7 seconds
- ■ One death every 13 seconds
- ■ One new immigrant every 45 seconds

Write a program to display the population for each of the next five years. Assume the current population is 312,032,486 and one year has 365 days. *Hint*: In Java, if two integers perform division, the result is an integer. The fractional part is truncated. For example, 5 / 4 is 1 (not 1.25) and 10 / 4 is 2 (not 2.5). To get an accurate result with the fractional part, one of the values involved in the division must be a number with a decimal point. For example, 5.0 / 4 is 1.25 and 10 / 4.0 is 2.5.

2 Write two overloaded methods that return the average of an array with the following headers:

```
public static int average(int[] array)
public static double average(double[] array)
```

Write a test program that random generate ten integers between 0 – 100, and ten double values between 0 – 50. Invokes both methods, and displays all values and the average values (All double numbers should displayed with two-digit decimal places).

3 **Test the program using the following points:**

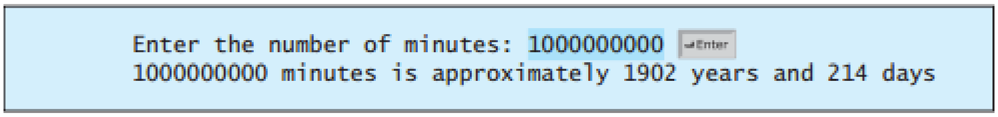
double[][] points = {{-1, 2, 3}, {-1, -1, -1}, {5, 2, 1}, {3, 0.5, 9}, {3.5, 2, -1.5}, {8, 1.5, - 3}, {-4, 4, 2}, {-7, 4, -0.5}};

The formula for computing the distance between two points (x1, y1, z1) and (x2, y2, z2) is square root of [(x1-x2)^2 + (y1-y2)^2 + (z1-z2)^2]. Sample output among 8 points:

{{-1, 2, 3}, {-1, -1, -1}, {5, 2, 1}, {3, 0.5, 9}, {3.5, 2, -1.5}, {8, 1.5, - 3}, {-4, 4, 2}, {-7, 4, -0.5}};

The closest two points are (x1, y1, z1) and (x2, y2, z2), and the distance is [min_distance]

4 (*Find the number of years*) Write a program that prompts the user to enter the minutes (e.g., 1 billion), and displays the number of years and days for the minutes. For simplicity, assume a year has 365 days. Here is a sample run:



```
Enter the number of minutes: 1000000000 ↵Enter
1000000000 minutes is approximately 1902 years and 214 days
```

5 **(*Phone keypad*):**

The international standard letter/number mapping found on the telephone is: (refer to the attachment)

Write a method that returns a number, given an uppercase letter, as follows:

public static int getNumber(char uppercaseLetter)

Write a test program that prompts the user to enter a phone number as a string. The input number may contain letters. The program translates a letter (upper- or lowercase) to a digit and leaves all other characters intact.

Sample output:

Enter a string: (input) **1-800-Flowers**

(output) **1-800-3569377**

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
*	0 +	#

6 The **Account** class has defined to model a bank account: An account has the properties account number, balance, annual interest rate, and date created, and methods to deposit and withdraw funds.

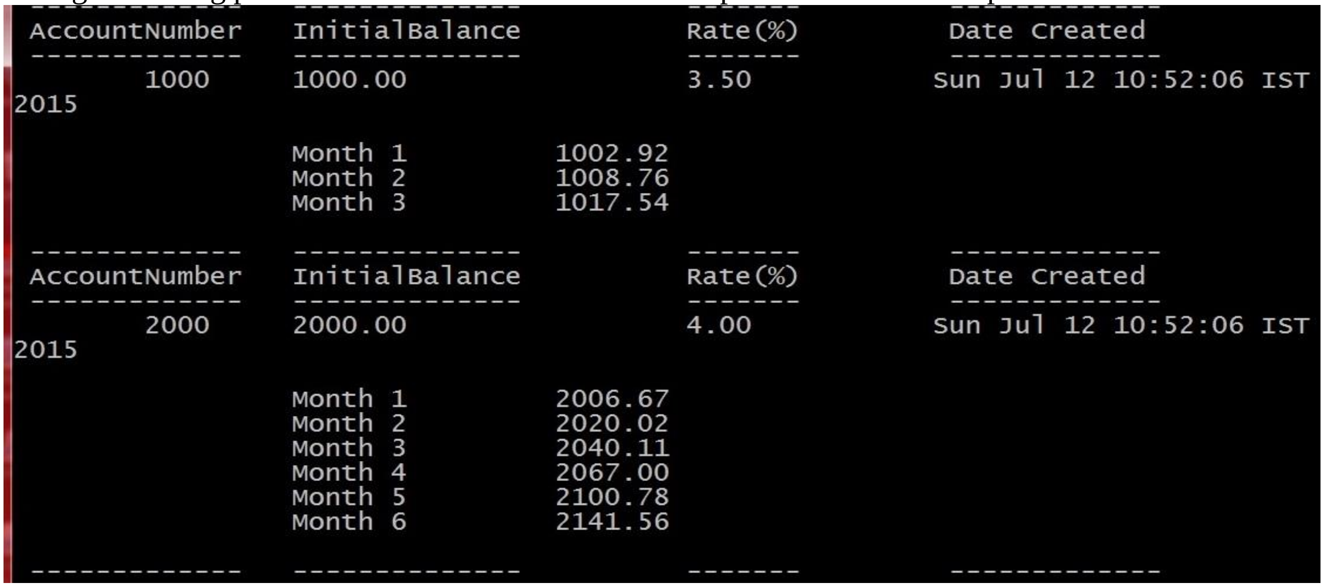
i. You will add two private variables, duration (in months) and CDannualInterestRate.

ii. Four methods in this class are getDuration(), setDuration(), getCDannualInterestRate(),and setCDannualInterestRate()

iii. Every three months CD duration will increase the CDannualInterestRate by 0.5% from annualInterestRate of the base class. For example: if annualInterestRate is 3%, the CDannualInterestRate will be 3.5% of a three-months CD saving account, and the CDannualInterestRate will be 4% of a six-month CD.

iv. Include **toString()** methods to print the account number, the annual interest rate, the initial balance, the monthly interest, and the date when this account was created.

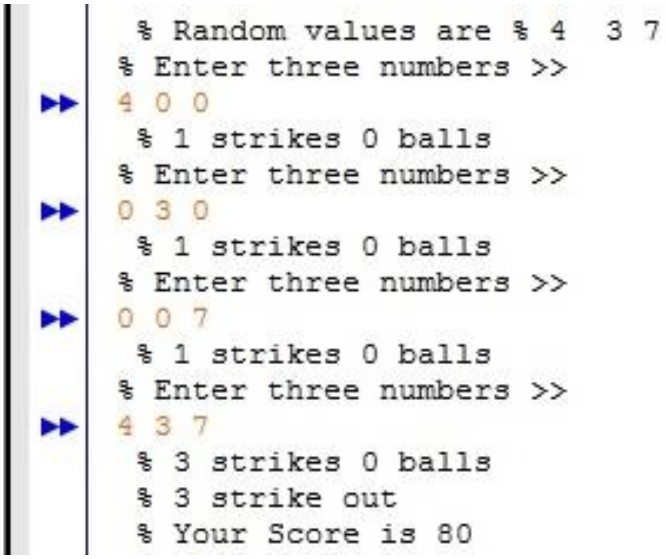
Write a driver program to test class CDAccount. Use ArrayList to create five objects of class CDAccount with balances of $1000.00 to $5000.00, and duration from 3-month to 15-months respectively. Set the annualInterestRate to 3%. Calculate and print the monthly interest of every month during the saving period for each CDAccount. Your output should be as sample screenshot shown below:



```
 -------------   --------------            -------          -------------
 AccountNumber   InitialBalance            Rate(%)          Date Created
 -------------   --------------            -------          -------------
      1000       1000.00                   3.50           Sun Jul 12 10:52:06 IST
2015

                 Month 1          1002.92
                 Month 2          1008.76
                 Month 3          1017.54

 -------------   --------------            -------          -------------
 AccountNumber   InitialBalance            Rate(%)          Date Created
 -------------   --------------            -------          -------------
      2000       2000.00                   4.00           Sun Jul 12 10:52:06 IST
2015

                 Month 1          2006.67
                 Month 2          2020.02
                 Month 3          2040.11
                 Month 4          2067.00
                 Month 5          2100.78
                 Month 6          2141.56

 ------------    --------------            -------          -------------
```

7 Purpose of the programming: You will write a baseball program. The user plays against the program. The program randomly chooses 3 different numbers. The user wins when he/she guesses those 3 numbers. The user enters the 3 numbers that he/she guesses; the program produces the total number of “strike” or “ball.” When the user correctly guesses both the number and its place, it is “strike”. When the user correctly guesses the number, but not its place, it is “ball”. The game is over with 3 strikes. At the end of the game, the program displays the final score. The final score is calculated by (100- (number of attempts*5).

Example: Say the program chose 4, 3, 7
Sample Screenshot of the program:



```
 % Random values are % 4  3 7
% Enter three numbers >>
4 0 0
 % 1 strikes 0 balls
% Enter three numbers >>
0 3 0
 % 1 strikes 0 balls
% Enter three numbers >>
0 0 7
 % 1 strikes 0 balls
% Enter three numbers >>
4 3 7
 % 3 strikes 0 balls
 % 3 strike out
 % Your Score is 80
```

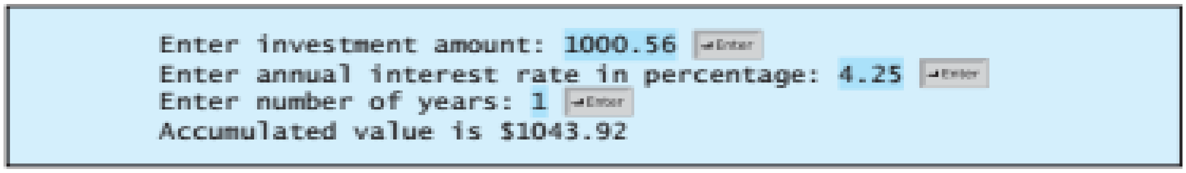
8 Generating a grid: Your program should fill the grid of size ‘n’ given by the user as input. Write a test program that fills all possible distinct numbers of random in that grid with the values of size (n*n) i.e., nXn values should be randomly filled in that gird. Your program should repeatedly display the output every time with distinct grids of values until user wants to quit or the user wants to give one more size for the grid. Your program should provide a menu with 3 options: 1 for take input for a grid 2. Generating grid values and displaying continuously with yes/no to display next grid with the same size of grid or he wants to stop. 3. Exit to quit from the program.

Sample output shown below:
Select the choice (1: Enter the grid size   2: Generate the grid   3: Exit): 1
You have selected option 1:
Please enter the size for the grid: 3
The grid with size 3x3 is:

4	**3**	**7**
1	2	5
6	9	8

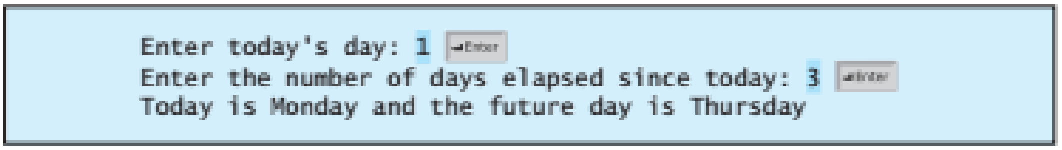
Do You want to continue (Y/N)? N
You have selected option No.
Select the choice (1: Enter the grid size   2: Generate the grid   3: Exit): 3
You have selected option 3.
Thank you!

9 (*Financial application: calculate future investment value*) Write a program that reads in investment amount, annual interest rate, and number of years, and displays the future investment value using the following formula:
futureInvestmentValue = investmentAmount * (1 + monthlyInterestRate)numberOfYears*12
For example, if you enter amount **1000.56**, annual interest rate **4.25%**, and number of years **1**, the future investment value is **1043.92**. Here is a sample run:

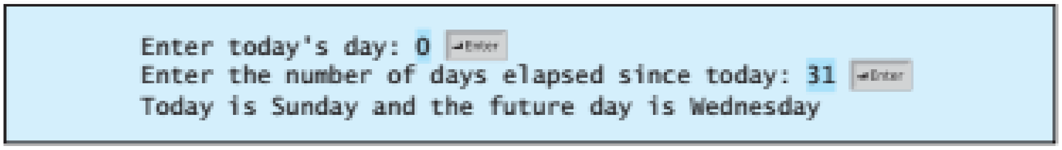


```
Enter investment amount: 1000.56 ↵Enter
Enter annual interest rate in percentage: 4.25 ↵Enter
Enter number of years: 1 ↵Enter
Accumulated value is $1043.92
```

10 (*Find future dates*) Write a program that prompts the user to enter an integer for today's day of the week (Sunday is 0, Monday is 1… and Saturday is 6). Also, prompt the user to enter the number of days after today for a future day and display the future day of the week. Here is a sample run:



```
Enter today's day: 1 ↵Enter
Enter the number of days elapsed since today: 3 ↵Enter
Today is Monday and the future day is Thursday
```

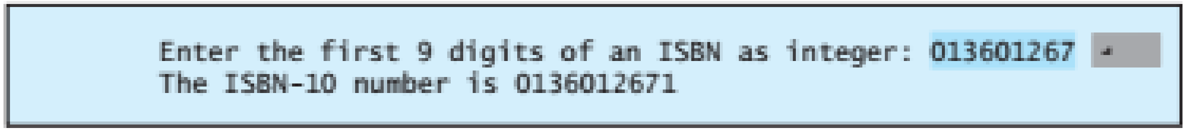


```
Enter today's day: 0 ↵Enter
Enter the number of days elapsed since today: 31 ↵Enter
Today is Sunday and the future day is Wednesday
```

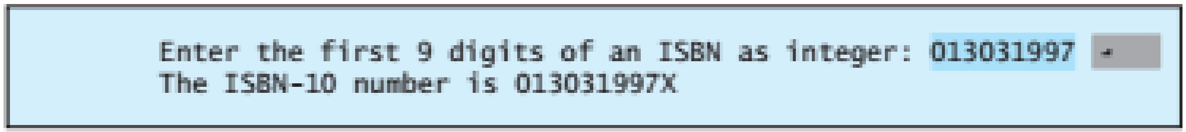
11 (*Business: check ISBN-10*) An **ISBN-10** (International Standard Book Number) consists of 10 digits: *d*1*d*2*d*3*d*4*d*5*d*6*d*7*d*8*d*9*d*10. The last digit, *d*10, is a checksum, which is calculated from the other nine digits using the following formula:

(*d*1 * 1 + *d*2 * 2 + *d*3 * 3 + *d*4 * 4 + *d*5 * 5 + *d*6 * 6 + *d*7 * 7 + *d*8 * 8 + *d*9 * 9) % 11

If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention. Write a program that prompts the user to enter the first 9 digits and displays the 10-digit ISBN (including leading zeros). Your program should read the input as an integer. Here are sample runs:

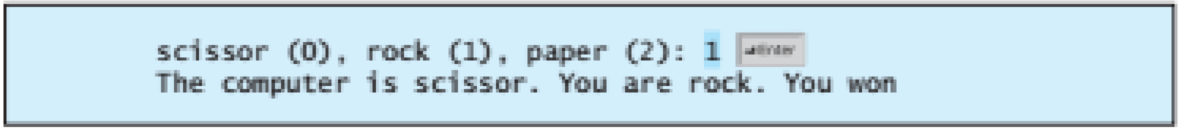


```
Enter the first 9 digits of an ISBN as integer: 013601267 ↵
The ISBN-10 number is 0136012671
```

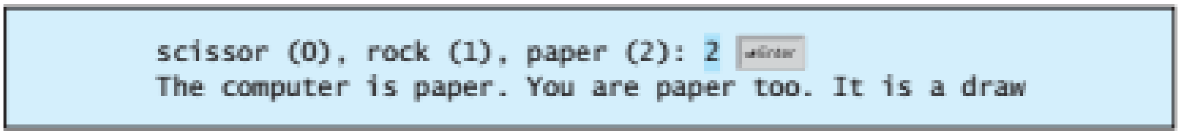


```
Enter the first 9 digits of an ISBN as integer: 013031997 ↵
The ISBN-10 number is 013031997X
```

12 (*Game: scissor, rock, paper*) Write a program that plays the popular scissor-rockpaper game. (A scissor can cut a paper, a rock can knock a scissor, and a paper can wrap a rock.) The program randomly generates a number 0, 1, or 2 representing scissor, rock, and paper. The program prompts the user to enter a number 0, 1, or 2 and displays a message indicating whether the user or the computer wins, loses, or draws. Here are sample runs:



```
scissor (0), rock (1), paper (2): 1 ↵Enter
The computer is scissor. You are rock. You won
```

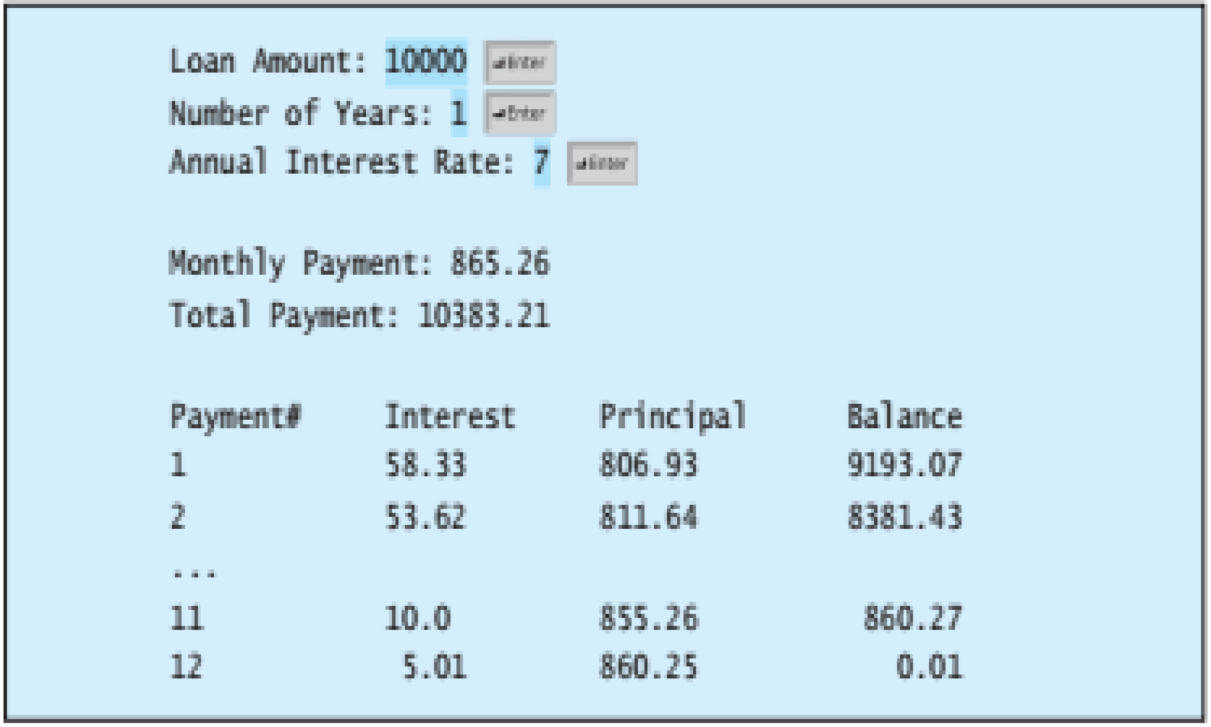


```
scissor (0), rock (1), paper (2): 2 ↵Enter
The computer is paper. You are paper too. It is a draw
```

13 (*Compute the perimeter of a triangle*) Write a program that reads three edges for a triangle and computes the perimeter if the input is valid. Otherwise, display that the input is invalid. The input is valid if the sum of every pair of two edges is greater than the remaining edge.

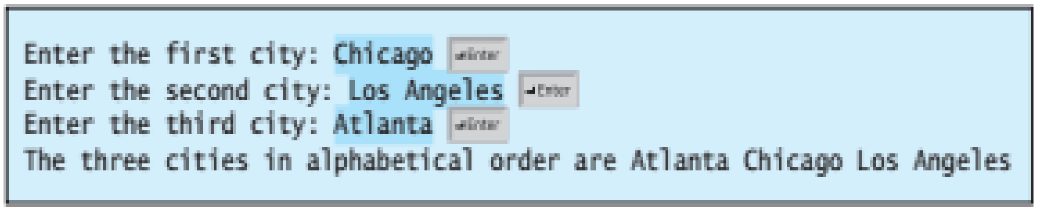
14 *(Financial application: loan amortization schedule)* The monthly payment for a given loan pays the principal and the interest. The monthly interest is computed by multiplying the monthly interest rate and the balance (the remaining principal). The principal paid for the month is therefore the monthly payment minus the monthly interest. Write a program that lets the user enter the loan amount, number of years, and interest rate and displays the amortization schedule for the
loan.

Here is a sample run:



```
Loan Amount: 10000 ↵Enter
Number of Years: 1 ↵Enter
Annual Interest Rate: 7 ↵Enter

Monthly Payment: 865.26
Total Payment: 10383.21

Payment#      Interest      Principal      Balance
1             58.33         806.93         9193.07
2             53.62         811.64         8381.43
...
11            10.0          855.26          860.27
12              5.01        860.25            0.01
```

15 (*Order three cities*) Write a program that prompts the user to enter three cities and displays them in ascending order. Here is a sample run:



```
Enter the first city: Chicago ↵Enter
Enter the second city: Los Angeles ↵Enter
Enter the third city: Atlanta ↵Enter
The three cities in alphabetical order are Atlanta Chicago Los Angeles
```