German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Assoc. Prof. Milad Ghantous
Assoc. Prof. Mohammed Hamed

Introduction to Computer Programming, Spring Term 2025
Practice Assignment 3

Exercise 3-1 Zodiac

Write a Java program that requests a month number (1-12) and a day number (1-31). The program should print the Zodiac Sign according to the user's input.

Sign	From	To
Capricorn	December 22	January 19
Aquarius	January 20	February 17
Pisces	February 18	March 19
Aries	March 20	April 19
Taurus	April 20	May 20
Gemini	May 21	June 20
Cancer	June 21	July 22
Leo	July 23	August 22
Virgo	August 23	September 22
Libra	September 23	October 22
Scorpio	October 23	November 21
Sagittarius	November 22	December 21

Solution:

```
import java.util.Scanner;

public class Zodiac {

    public static void main(String [] args) {

        int month, day;
        String horoscope;
        Scanner sc = new Scanner(System.in);

        System.out.print("Please␣enter␣the␣month(1-12):␣");
        month = sc.nextInt();
        System.out.print("Please␣enter␣the␣day(1-31):␣");
        day = sc.nextInt();

        switch (month) {
        case 1: horoscope = (day <= 19) ? "Capricorn" : "Aquarius";
                break;
        case 2: horoscope = (day <= 17) ? "Aquarius" : "Pisces";
                break;
        case 3: horoscope = (day <= 19) ? "Pisces": "Aries";
                break;
        case 4: horoscope = (day <= 19) ? "Aries" : "Taurus";
                break;
        case 5: horoscope = (day <= 20) ? "Taurus" : "Gemini";
                break;
        case 6: horoscope = (day <= 20) ? "Gemini" : "Cancer";
                break;
        case 7: horoscope = (day <= 22) ? "Cancer" : "Leo";
                break;
        case 8: horoscope = (day <= 22) ? "Leo" : "Virgo";
                break;
        case 9: horoscope = (day <= 22) ? "Virgo": "Libra";
                break;
        case 10: horoscope = (day <= 22) ? "Libra": "Scorpio";
                break;
        case 11: horoscope = (day <= 21) ? "Scorpio" : "Sagittarius";
                break;
        case 12: horoscope = (day <= 21) ? "Sagittarius" : "Capricorn";
                break;
        default: horoscope = "The␣value␣you␣entered␣for␣the␣month␣is␣not␣correct!";
                break;
        }
        System.out.print(horoscope);
    }

}
```

Exercise 3-2 ATM

Write a program that asks a user to enter his/her pincode to the ATM machine. The pincode is generated randomly once and user is notified with this pincode. The ATM machine then asks the user to enter the pincode and the user has maximum 3 trials to enter. After that, a corresponding message should be shown to the user. If the user succeeds to get the pincode, a message with the number of trials should be displayed.

The output should be something like:

```
Please enter your pincode
5643

Please enter your pincode
5645

Please enter your pincode
5465

You exceeded your trials. We will lock your account!

Please enter your pincode
6523

Please enter your pincode
9654

You succeeded to guess your pincode in trial number 2
```

Solution:

- Using while

```
import java.util.Scanner;
import java.util.Random;

public class Atm {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        Random r = new Random();

        int otp = r.nextInt(10000-1000)+1000;
        System.out.println("OTP: " + otp);

        System.out.print("Enter your pincode: ");
        int pinCode = sc.nextInt();

        int i=1;
        while(otp!=pinCode && i<3){
                System.out.print("Enter your pincode: ");
                pinCode=sc.nextInt();
                i++;
        }

        if(pinCode==otp)
         System.out.println("You succeeded in trial number: " + (i));
        else
         System.out.println("You exceeded your trials , we'll lock ur account!");
    }}
```

- Using for

```
import java.util.Scanner;
import java.util.Random;

public class Atm {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Random r = new Random();

        int otp = r.nextInt(10000-1000)+1000;
        System.out.println("Your␣OTP:␣" + otp);
        String res = "You␣exceeded␣your␣trials.␣We␣will␣lock␣your␣account!";

        int pinCode;
        for(int i=1; i<=3; i++){
        System.out.print("Trial:␣" + i+ ",␣Enter␣your␣code:␣");
        pinCode = sc.nextInt();
        if(otp==pinCode){
            res= "You␣succeeded␣to␣guess␣your␣pincode␣in␣trial␣number:␣" + i;
            break;
        }
    }
    System.out.print(res);}}
```

Exercise 3-3 Caesar Cipher

Write a Java program which takes two input variables `message` of data type `String` and `key` of data type `int`. The program should shift each character in `message` with a distance of `key`. For example: if `key=3` then `a` will be replaced by `d` and `b` will be replaced by `e` and so on.

Hint: You can use the following method

- `charAt(int index)`: Returns the character at the specified index. The first character of the sequence is at index 0, the next at index 1 and so on.

```
String s = "Hello";
char   c = s.charAt(0);
```

The value of `c` is `'H'`.

The output should be somthing like this:

```
Please enter the Message:
Hat
Please Enter the Key:
3
The encrypted word is:
Kdw
```

Solution:

- Using while

```
import java.util.*;
public class Caesar
{
 public static void main(String[]args)
 {
        Scanner sc = new Scanner (System.in);
        System.out.println("Please␣Enter␣a␣Word␣to␣be␣Encrypted:␣");
        String s = sc.nextLine();
        System.out.println("Please␣Enter␣a␣Key:␣");
        int key = sc.nextInt();
        int ascii = 0;
        char x;
        int len = s.length();
        int i = 0;
        while (i < len)
        {
                ascii = s.charAt(i)+key;
                if((ascii >'z' && s.charAt(i)>= 'a' && s.charAt(i)<'z')
                || (ascii >'Z' && s.charAt(i)>= 'A' && s.charAt(i)<'Z'))
                        ascii -= 26;
                x = (char) ascii;
                System.out.print(x);
                i++;
        }
        System.out.println();
 }
}
```

- Using for

```
import java.util.*;
public class Caesar {
    public static void main(String[] args)
    {
      Scanner sc = new Scanner (System.in);
      System.out.println("Please␣Enter␣a␣Word␣to␣be␣Encrypted:␣");
      String  s  = sc.nextLine();
      System.out.println("Please␣Enter␣a␣Key:␣");
      int    key = sc.nextInt();
      int ascii = 0;
      char x;
      int len = s.length();
      for (int i = 0;i<len;i++)
      {
        ascii =  s.charAt(i) + key;
        if((ascii>'z' && s.charAt(i)>='a' && s.charAt(i)<='z')
             || (ascii>'Z' && s.charAt(i)>='A' && s.charAt(i)<='Z'))
           ascii -= 26;
        x = (char) ascii;
        System.out.print(x);
      }
        System.out.println();
    }
}
```

Exercise 3-4 String Manipulation I

Write a program that determines the number of consonants, vowels, punctuation characters, and spaces in an input line. Read in the line into a String (in the usual way). Now use the `charAt()` method in a loop to access the characters one by one. Use a switch statement to increment the appropriate variables based on the current character. After processing the line, print out the results.
Hint: Use the .toLowerCase() method to make all letters in lower case.

Solution:

```
import java.util.*;
public class Chars
{
   public static void main (String[] args)
   {
      Scanner sc = new Scanner (System.in);
      int consonant = 0, vowel = 0, punctuation = 0, space = 0, digit = 0;
      String str, lowered;
      System.out.print("Enter␣a␣string:␣");
      str     = sc.nextLine();

      //Convert the letter of the String 'str' to lower case.
      lowered = str.toLowerCase();

      for (int i = 0; i<lowered.length(); i++){

        switch(lowered.charAt(i)){
          case '␣': space++; break;
          case 'a':
          case 'e':
```

```
            case 'i':
            case 'o':
            case 'u': vowel++; break;
            case 'b':
            case 'c':
            case 'd':
            case 'f':
            case 'g':
            case 'h':
            case 'j':
            case 'k':
            case 'l':
            case 'm':
            case 'n':
            case 'p':
            case 'q':
            case 'r':
            case 's':
            case 't':
            case 'v':
            case 'w':
            case 'x':
            case 'y':
            case 'z': consonant++; break;
            case '0':
            case '1':
            case '2':
            case '3':
            case '4':
            case '5':
            case '6':
            case '7':
            case '8':
            case '9': digit++; break;
            default:  punctuation++;
          }
      }
      System.out.println ("Number of consonants: " + consonant);
      System.out.println ("Number of vowels: " + vowel);
      System.out.println ("Number of digits: " + digit);
      System.out.println ("Number of punctuations: " + punctuation);
      System.out.println ("Number of spaces: " + space);
    }
}
```

Exercise 3-5 Score

A sequence of six tests, all scored out of 100, are to be given different weightings in determining a final mark. Write a Java program that computes the appropriate weighted score for one test. The fragment should first read values of testNumber and score. Using a switch statement, it should then compute and print the appropriate value of weightedScore using the weightings given in the following table.

Test Number	Weight
1	10%
2	20 %
3	20 %
4	15%
5	15%
6	20%

Solution:

```
import java.util.Scanner;

public class Score {

    public static void main(String [] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("please enter the test number and the score");
        int testNumber = sc.nextInt();
        int score = sc.nextInt();
        double weight = 0;
        switch (testNumber) {
            case 1:
            weight = 0.1 * score;
            break;
            case 2:
            case 3:
            case 6:
            weight = 0.2 * score;
            break;
            case 4:
            case 5:
            weight = 0.15 * score;
            break;
            default:
            weight = 0;
        }
        System.out.println("A score of " + score+" on test "+testNumber+" gives a
        weighted score of "+weight +".");
    }
}
```

Exercise 3-6 Fixed Length

Write a program that asks the user to enter two words. The program then prints out both words on one line. The words will be separated by enough dots so that the total line length is 30. We can use it to make an index for a book. The user enters the name of the chapters/sections and the page number and the program generate the index. You can only print one dot at a time.

```
Enter first word:
Chapter 5
Enter second word:
153

Chapter 5...................153
```

Solution:

```
import java.util.*;
public class Word {
   public static void main (String[] args)
   {
     Scanner sc = new Scanner (System.in);
     String word1, word2, line = "";
     int dots;

     System.out.print ("Enter␣first␣word:␣");
     word1 = sc.nextLine();
     System.out.print ("Enter␣second␣word:␣");
     word2 = sc.nextLine();

     dots  = 30 - (word1.length() + word2.length());
     line  = line.concat(word1);

     for (int i = 0; i<dots; i++)
        line = line.concat(".");

     line = line.concat(word2);
     System.out.println (line);
   }
}
```

Exercise 3-7 Stream of Numbers

Write a Java program to read a list of nonnegative integers and outputs the maximum integer, the minimum integer, and the average of all the integers. The end of the input is indicated by the user entering a negative number. Note that the negative number is not used in finding the maximum, minimum, or average. The output should be something like this:

```
Please enter a sequence of positive numbers
2
3
5
4
-1
The maximum number is : 5
The minimum number is: 2
The average is: 3.5
```

Solution:

- Using While

```
import java.util.*;
public class NumbersWhile {
  public static void main(String[] args)
  {
    Scanner sc = new Scanner (System.in);
    System.out.println("Please␣enter␣the␣number");
    int num    = sc.nextInt();
    if(num<0)
       System.out.println("No␣positive␣Numbers␣entered");
    else {
      int small, large;
      small = num;
      large = num;
      double sum = 0;
      double avg;
      int count  = 0;

      while (num>=0)
      {
        if(num<small)
          small = um;
        else if (num>large)
          large = num;

        sum += num;
        count++;
        System.out.println("Please␣enter␣another␣number:");
        num = sc.nextInt();
      }
      avg = sum/count;
      System.out.println("The␣average␣of␣the␣numbers␣is␣␣␣␣␣␣" + avg);
      System.out.println("The␣smallest␣integer␣you␣entered␣is␣" + small);
      System.out.println("The␣largest␣integer␣you␣entered␣is␣␣" + large);
    }
  }
}
```

Exercise 3-8 Months

Write a Java program that prints the number of days for any given month.

```
Please enter the month number (1-12): 5
31 days.
```

Solution:

```
import java.util.Scanner;

public class Month {

    public static void main(String [] args) {

        int month;
        String numberofdays;
        Scanner sc = new Scanner(System.in);
        System.out.print("Please enter the month number (1-12): ");
        month = sc.nextInt();
        switch (month) {
        case 1:
            /* falls through */
        case 3:
            /* falls through */
        case 5:
            /* falls through */
        case 7:
            /* falls through */
        case 8:
            /* falls through */
        case 10:
            /* falls through */
        case 12:
            numberofdays = "31 days.";
            break;
        case 4:
            /* falls through */
        case 6:
            /* falls through */
        case 9:
            /* falls through */
        case 11:
            numberofdays = "30 days.";
            break;
        case 2:
            numberofdays = "either 28 or 29 days.";
            break;
            numberofdays = "The value you entered for the month is not correct!";
            break;
        }
        System.out.println(numberofdays);
    }
}
```

Exercise 3-9 Adder

Write a program that adds up integers that the user enters. First the programs asks how many numbers will be added up. Then the program prompts the user for each number. Finally, it prints the sum.

The output should be something like:

```
How many integers will be added:
5
Enter integer 1:
3
Enter integer 2:
4
Enter integer 3:
-4
Enter integer 4:
-3
Enter integer 5:
7

The sum is 7
```

Solution:

```
import java.util.*;
public class Adding {
  public static void main (String[] args)
  {
    Scanner sc = new Scanner (System.in);

    System.out.println("How␣many␣integers␣will␣be␣added?");
    int n = sc.nextInt();
    int i, x, sum = 0;

    for(i = 0; i<n ; i++)
    {
       System.out.println("Enter␣integer␣"+(i+1));
       x = sc.nextInt()
       sum += x;
    }
    System.out.println("The␣sum=␣"+sum);
  }
}
```

Exercise 3-10 Euclidean Algorithm

The Euclidean algorithm determines the greatest common divisor (GCD) of two positive numbers by repeatedly replacing the larger number with the result of subtracting the smaller one from it until the two numbers are equal.

Write a Java program for Euclidean algorithm where the user has to enter the two numbers and the program should calculate their greatest common divisor. The output should be something like:

```
Please, enter a first number:
45
Please, enter a second number:
22
The GCD of 45 and 22 is 1
```

Solution:

- Using while

```
import java.util.*;
public class Euclidian {
    public static void main(String[] args)
    {
      Scanner sc = new Scanner (System.in);
      System.out.println("Please, enter a number:");
      int num1 = sc.nextInt();

      System.out.println("Please, enter a second number:");
      int num2 = sc.nextInt();

      int num1Saved = num1;
      int num2Saved = num2;

      while(num1 != num2)
      {
        if(num1 > num2)
            num1 -= num2;
        else
            num2 -= num1;
      }
      System.out.println("The GCD of " + num1Saved + " and " + num2Saved +
                         " is " + num1);
    }
}
```

- Using for

```
import java.util.*;
public class Euclidean
{
 public static void main(String[]args)
 {
        Scanner sc = new Scanner(System.in);
        System.out.println("Please Enter a number: ");
        int num1 = sc.nextInt();

        System.out.println("Please Enter a second number: ");
        int num2 = sc.nextInt();

        int num1Saved = num1;
        int num2Saved = num2;

        for(;num1!=num2;)
        {
                if(num1 > num2)
                        num1 -= num2;
                else
                        num2 -= num1;
        }
        System.out.println("the GCD of " + num1Saved + " and " + num2Saved +
                           " is "+ num1);
 }
}
```

Exercise 3-11 String Manipulation II

Write a program that given two strings, checks if they are equal or not.
Hint You are not allowed to use .equals method.

Solution:

```
import java.util.Scanner;
public class CheckEquality
{
        public static void main(String[]args)
        {
                Scanner sc = new Scanner (System.in);
                System.out.println("Enter␣a␣word");
                String s1 = sc.next();
                System.out.println("Enter␣another␣word");
                String s2 = sc.next();
                boolean flag = true;
                if(s1.length()!=s2.length())
                        System.out.println("Strings␣do␣not␣match");
                else
                {
                        for(int i = 0; i < s1.length() && flag == true; i++)
                        {
                                if(s1.charAt(i)!=s2.charAt(i))
                                        flag = false;
                        }
                if(flag)
                        System.out.println("Strings␣match");
                else
                        System.out.println("Strings␣do␣not␣match");
                }
        }
}
```

Exercise 3-12 String Manipulation III

Write a java program that determines whether the text the user inputs is a **palindrome** or not. A palindrome is a piece of text that can be read the same way in either direction (left to right and right to left). Examples of palindromes include words such as `racecar` and `noon`.

Solution:

```
import java.util.Scanner;
public class Palindrome
{
 public static void main(String[]args)
 {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter␣a␣word");
        String word = sc.next();
        boolean flag = true;
        String reverse = "";
        for(int i= word.length()-1; i >=0;i--)
                reverse += word.charAt(i);
        if(word.equals(reverse))
                System.out.println("the␣word␣" + word + "␣is␣a␣palindrome");
        else
                System.out.println("the␣word␣" + word + "␣is␣not␣a␣palindrome");
 }
}
```