Kingdom of Saudi Arabia
Ministry of Education
Taibah University
College of Computer Science &
Engineering
Department of Computer Science

المملكة العربية السعودية
وزارة التعليم
جامعة طيبة

كلية علوم وهندسة حاسب الآلي
قسم علوم الحاسب الآلي

Date:

Lab Manual

Programming 1

CS 111

Student ID:

Student Name:

January 2019

Acknowledgment

I would like to thank Dr. Abdullah Alsaeedi for his idea of preparing a lab manual that could help improving the programming skills of CS students.

Also, I would like to thank the following members of the lab manual committee for the high level of professionalism and dedication they had shown in preparing and writing the lab manual:

- Dr. Mutaz Abusara
- Dr. Hamza Ghandorh
- Dr. Reyadh Alluhaibi
- Mrs. Modhi AlSobeihy
- Mr. Raja Al-hejaili

Chair of lab manual committee

Dr. Barzaiq

January 2019

Declaration

The resources used to prepare and write the lab manual are:

- The academic experience of the members of the lab manual committee.

- The textbook of Y. Daniel Liang, titled "Introduction to Java programming", tenth edition.

Lab Instructions

- You are not allowed to attend the lab without your lab manual.

- Do not mess with the lab computers or accessories.

- Do not mess with software installed on computers.

- Food and beverages are not allowed in the lab.

- No obnoxious or belligerent behavior will be tolerated.

- Turn off your mobile phone or put it on silent mode.

ASSIGNMENT DECLARATION FORM

(Group Assignment)

❖ **By submitting this work and signing this document, we declare that:**

1. This assignment is our own original work.

2. No part of this work has been copied from any other source or person except where explicitly stated otherwise by reference or acknowledgment.

3. No part of this work has been previously submitted for assessment at this or any other institution.

4. We may be subject to student discipline processes in the event of an act of academic misconduct by us including an act of plagiarism or cheating.

Course Code	Course Title

Student ID	Student Name	Signature of student
Student ID	**Student Name**	**Signature of student**
Student ID	**Student Name**	**Signature of student**
Date of signature		

ASSIGNMENT DECLARATION FORM

(Individual Assignment)

❖ **By submitting this work and signing this document, I declare that:**

1. This assignment is my own original work.

2. No part of this work has been copied from any other source or person except where explicitly stated otherwise by reference or acknowledgment.

3. No part of this work has been previously submitted for assessment at this or any other institution.

4. I may be subject to student discipline processes in the event of an act of academic misconduct by me including an act of plagiarism or cheating.

Course Code	
Course Title	
Student ID	
Student Name:	
Signature of student	
Date of signature	

Table of Contents

Lab 01
Algorithms, Pseudocodes and Flowcharts – Week 01

Objectives*:*

- To solve a problem by using a pseudocode.
- To solve a problem by using an algorithm.
- To graphically represent the sequence of operations of an algorithm by using a flowchart.

Examples:

Example01. Please, do the following:

a-) Write a pseudocode to convert a temperature from Fahrenheit (℉) to Celsius (℃)

- Input a temperature in Fahrenheit

- Calculate temperature with formula C=5/9*(F-32)

- Print C

---

b-) Write an algorithm for the same problem

Step 1: Input F
Step 2: C ← 5/9 * (F-32)
Step 3: Print C

---

c-) Draw a flowchart for the same problem

```
        ┌─────────────────┐
        (     START       )
        └────────┬────────┘
                 │
                 ▼
        ╱─────────────────╱
       ╱     Input        ╱
      ╱        F         ╱
     ╱─────────────────╱
                 │
                 ▼
     ┌─────────────────────┐
     │  C ← 5/9 * (F-32)   │
     └──────────┬──────────┘
                │
                ▼
       ╱──────────────────┐
      ╱       Print        │
      ╲         C          │
       ╲──────────────────┘
                │
                ▼
        ┌─────────────────┐
        (      END        )
        └─────────────────┘
```
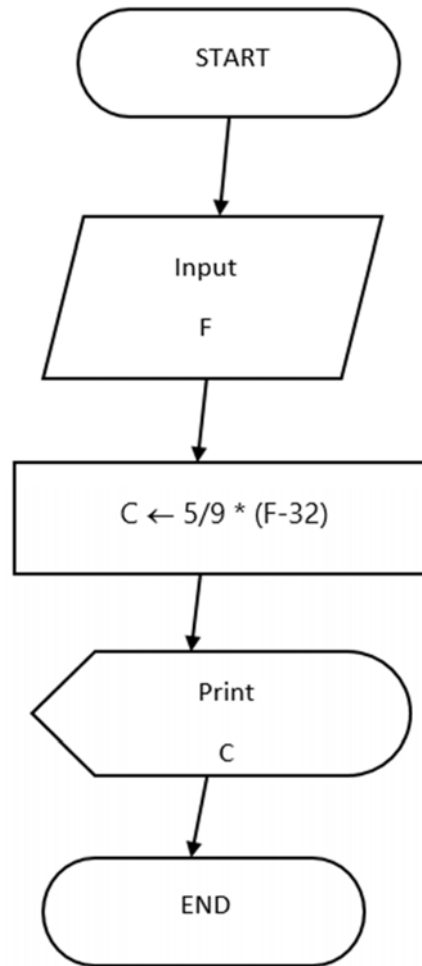
Exercises:

Exercise01. Please, do the following:

a-) Write a pseudocode to read three numbers and find the smallest one

Input three numbers v1,v2,v3
compare v1 with v2 with v3, if v1 is smallest
then
the smallest (s) is v1
else
compare v1 with v2 with v3, if v2 is smallest
then
the smallest (s) is v2
else
the smallest (s) is v3
endif
print the smallest (s)

Input three numbers v1,v 2, v3
Compare v1 with v2, if v1 is smaller that v 2
Then
Compare v1 with v3, if v1 is smaller that v 3
Then
The smallest (s) is v1
Else
The smallest (s) is v3

Else
Compare v2 with v3, if v2 is smaller that v 3
Then
The smallest (s) is v2
Else
The smallest (s) is v3

endIf
Print the smallest (s)

b-) Write an algorithm for the same problem

c-) Draw a flowchart for the same problem

Homework:

Homework01. Please do the following:

a-) Write a pseudocode that receives a positive number from the user, and then, starts printing and keep printing the word "CS" until the number of the printings matches the received number from the user.

b-) Write an algorithm for the same problem.

c-) Draw a flowchart for the same problem.

input number (n) greater than or equal zero
print the word''cs'' ,if the printing equal the number (n)
then
stop printing
else
keep printing until the printings equal the number(n)
endif

step1: input (n)
step2: print "cs"
step3:if printing=(n),then
stop
else
keep printing until=(n)
endif

start
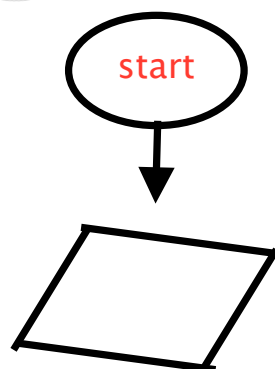
نص

Lab 02
Chapter 01 Introduction to Java – Week 02

Objectives:
- To provide a simple and quick introduction to the NetBeans IDE
- To write simple Java programs using NetBeans.

Downloading and Installing Java JDK with NetBeans IDE:
- Please, use the following link to download and then install Java JDK and NetBeans IDE:

  https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html

Setting up a Java project:
1. Start NetBeans IDE
2. In the IDE, choose **File > New Project**

File	Edit	View	Navigate	Source	Refactor	Run	Debug

New Project...	Ctrl+Shift+N	
New File...	Ctrl+N	
Open Project...	Ctrl+Shift+O	
Open Recent Project	▶	
Close Project (MobileApplication)		
Open File...		
Open Recent File	▶	
Project Group	▶	
Project Properties (MobileApplication)		
Import Project	▶	
Export Project	▶	
Save	Ctrl+S	
Save As...		
Save All	Ctrl+Shift+S	
Page Setup...		
Print...	Ctrl+Alt+Shift+P	
Print to HTML...		
Exit		

3. In the **New Project** wizard, expand the Java category and select **Java Application**, and then, click **Next**
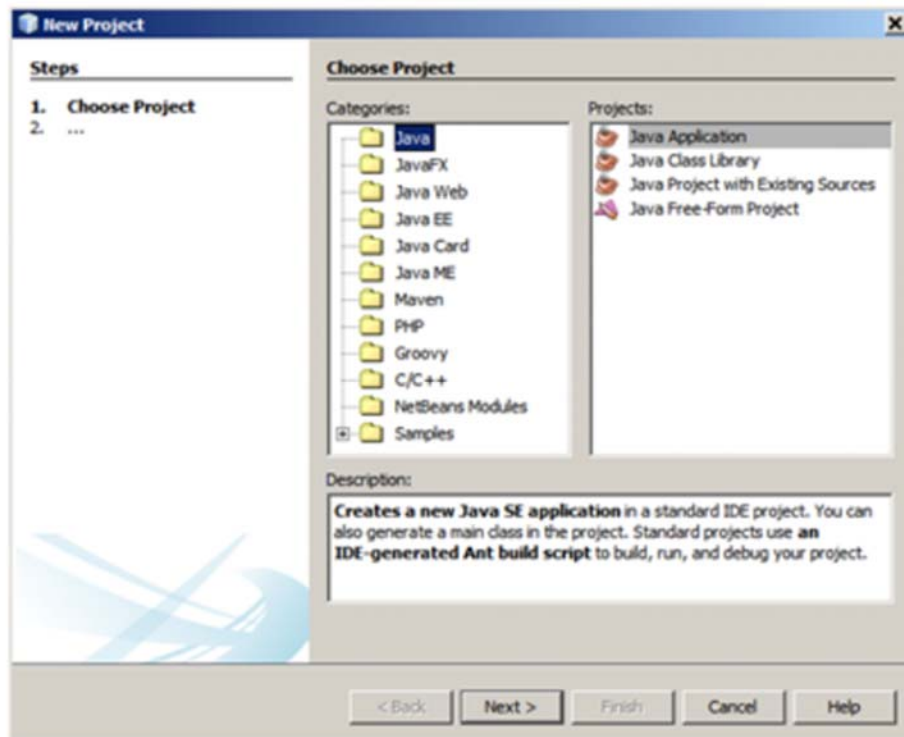
4. In the **Name and Location** page of the wizard, do the following:

    4.a In the **Project Name** field, type the name of your project. For example "**HelloWorld**"

    4.b In the **Create Main Class** field, type "**helloworld.HelloWorld**"
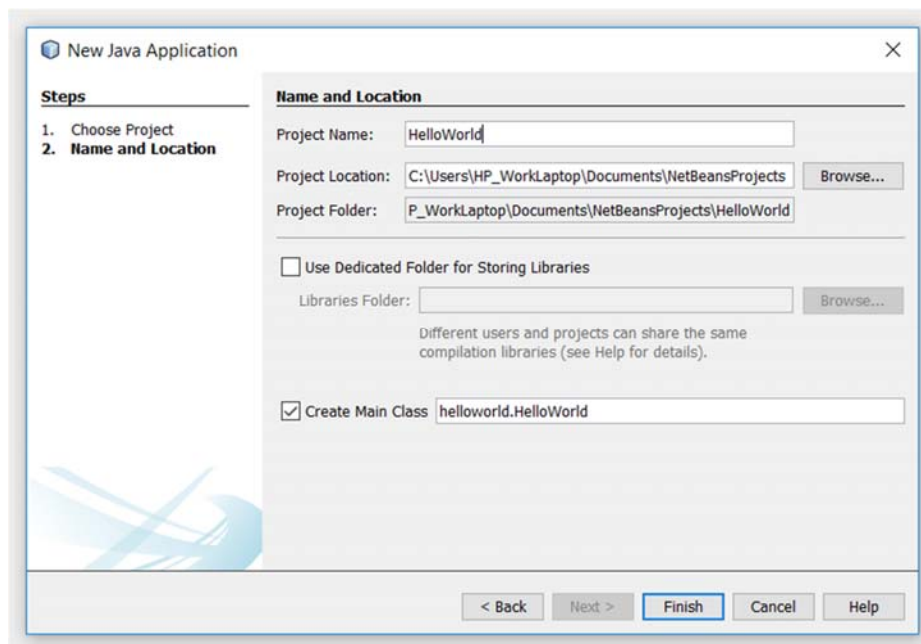
5. Click **Finish**

6. Now, after creating your first Java project, you should see the following components:

   6.a The **Project** window that contains a tree-view of the components of your project

   6.b The **Source Editor** window with a file called **HelloWorld** open

   6.c The **Navigator** window that you can use to quickly navigate between elements within the selected class

System.out.println ("Ruba");
System.out.println ("Ali");
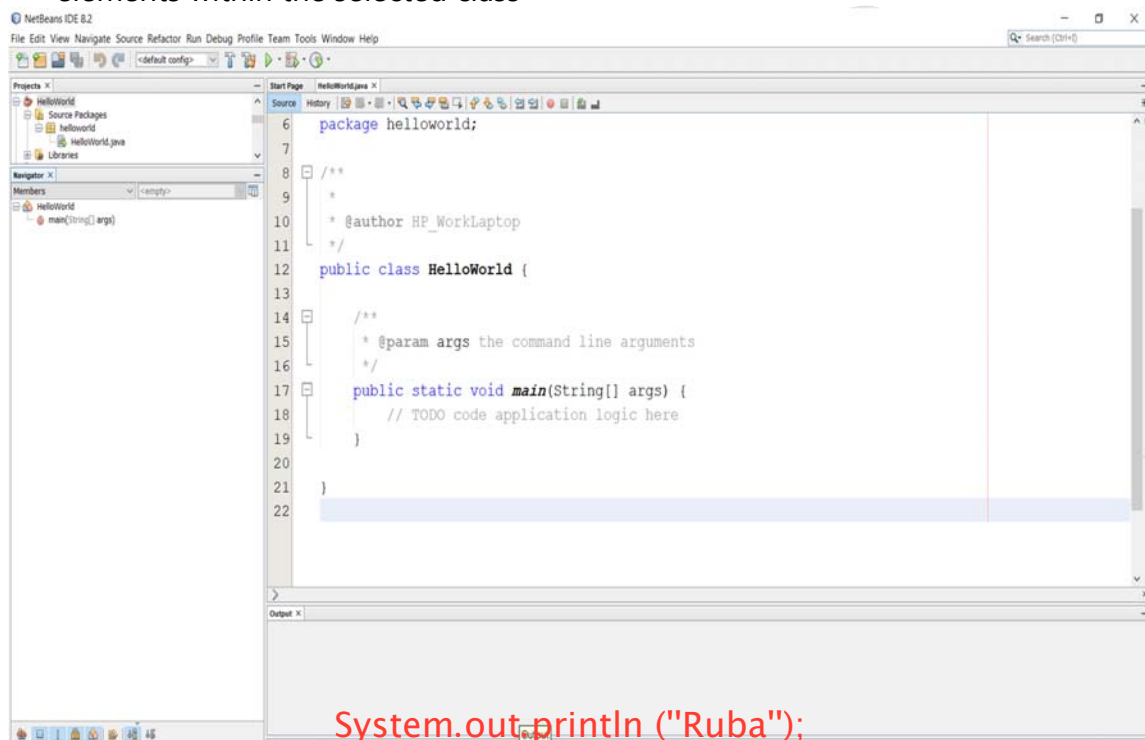
Examples:

Example01. Write the following Java program:

| ```java
public class HelloWorld{
  public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
``` | Output:<br>Hello World! |

Example02. Write the following Java program:

| ```java
public class Ch01_Week02_E2{
  public static void main(String[] args)
    {
        System.out.println("Programming is fun!");

        System.out.println("Fundamentals First");

        System.out.println("Problem Driven");

    }
}
``` | Output:<br>Programming is fun!<br>Fundamentals First<br>Problem Driven |

Example03. Write the following Java program:

| ```java
public class Ch01_Week02_E3{
  public static void main(String[] args)
  {
    System.out.print("(10.5 + 2 * 3) / (45 – 3.5) = ");

    System.out.println((10.5 + 2 * 3) / (45 - 3.5));

  }
}
``` | Output:<br>(10.5 + 2 * 3) / (45 – 3.5) =<br>0.39759036144578314 |

Exercises:

Exercise01. Please, do the following:

a-) Use the table below to write a Java program to print your first name and your family name

b-) Write the same program using NetBeans

Exercise02. Please, do the following:

a-) Use the table below to write a Java program to print your age, your weight and the sum of both values

b-) Write the same program using NetBeans

Homework:

Homework01. Write on a paper a Java program that prints

a-) the names of your family members

b-) their ages

c-) the sum of their ages

Lab 03
Chapter 02 Elementary Programming – Week 03

Objectives:

- To use different data types with variables and constants.
- To use different numeric expressions and operations.
- To obtain input from the console using the Scanner class

Examples:

Example01. Write a Java program that converts a Fahrenheit degree to Celsius

	Output:
| ```java
import java.util.Scanner;

public class Ch02_Week03_E1{
  public static void main(String[] args)
   {
      Scanner input = new Scanner(System.in);

     System.out.print("Enter a degree in Fahrenheit: ");
     double fahrenheit = input.nextDouble();

     // Convert Fahrenheit to Celsius
    double celsius = (5.0 / 9) * (fahrenheit - 32);
    System.out.println("Fahrenheit \"" + fahrenheit
    + "\" is \"" + celsius + "\" in Celsius");
    }
}
``` | Enter a degree in Fahrenheit: 32<br>Fahrenheit "32.0" is "0.0" in Celsius |

Example02. Write a Java program that displays the current time in GMT (Greenwich Mean Time) in the format hour : minute : second, such as 13:19:8.

| ```java
public class Ch02_Week03_E2{
  public static void main(String[] args)
    {
 // Obtain the total milliseconds since midnight, Jan 1, 1970
    long totalMilliseconds = System.currentTimeMillis();

    // Obtain the total seconds since midnight, Jan 1, 1970
    long totalSeconds = totalMilliseconds / 1000;

    // Compute the current second in the minute in the hour
    long currentSecond = totalSeconds % 60;

    // Obtain the total minutes
    long totalMinutes = totalSeconds / 60;

    // Compute the current minute in the hour
    long currentMinute = totalMinutes % 60;

    // Obtain the total hours
    long totalHours = totalMinutes / 60;

    // Compute the current hour
    long currentHour = totalHours % 24;

    // Display results
    System.out.println("Current time is " + currentHour + ":"
     + currentMinute + ":" + currentSecond + " GMT");
    }
}
``` | Output:<br>Current time is<br>12:38:28 GMT |

Exercises:

Exercise01. Write a Java program that reads an integer between 0 and 1000 and adds all the digits in the integer. For example, if an integer is 932, the sum of all its digits is 14. Hint: Use the % operator to extract digits, and use the / operator to remove the extracted digit. For instance, 932 % 10 = 2 and 932 / 10 = 93.
Here is a sample run:

```
Enter a number between 0 and 1000: 999  ↵ Enter
The sum of the digits is 27
```

Exercise02. Given an airplane's acceleration **a** and take-off speed **v**, you can compute the minimum runway length needed for an airplane to take off using the following formula:

$$length = \frac{v^2}{2a}$$

Write a Java program that prompts the user to enter **v** in meters/second (m/s) and the acceleration **a** in meters/second squared (m/s2), and displays the minimum runway length. Here is a sample run:

```
Enter speed and acceleration: 60 3.5  ↵Enter
The minimum runway length for this airplane is 514.286
```

Homework:

Homework01. Write a Java program that prompts the user to enter the seconds as a **String** (e.g., "1500500500"), converts the received value to **long**, and displays the number of years and days for the received seconds. For simplicity, assume a year has 365 days. Here is a sample run:

```
Enter the number of seconds: 1500500500
'1500500500' seconds is approximately '4' years and '2766' days
```

Lab 04
Chapter 02 Elementary Programming – Week 04

Objectives*:*

- To use Math class.
- To use different numeric expressions and operations.
- To manipulate string variables.

Examples:

Example01. Write a Java program that calculates the amount of paint needed to paint the walls and the ceiling of a room. Your program should ask the user to provide the values of **length**, **width**, and **height** of the room. Assuming that the room has doors and windows but they don't need painting. Also the floor in the room will not be painted. Ask the user to enter the number of doors and number of windows in the room, and adjust the total square feet to be painted accordingly. Assume that each door is 20 square feet and each window is 15 square feet. Suppose the paint covers 350 square feet per gallon.

```java
import java.util.Scanner;

public class Ch02_Week04_E1{
  public static void main(String[] args)
    {
        double roomLength, roomWidth, roomHeight;
        int roomDoors, roomWindows;

        Scanner input = new Scanner(System.in);

        System.out.print("Enter the length: ");
        roomLength = input.nextDouble();

        System.out.print("Enter the width: ");
        roomWidth = input.nextDouble();

        System.out.print("Enter the height: ");
        roomHeight = input.nextDouble();

        System.out.print("Enter the number of doors: ");
        roomDoors = input.nextInt();

        System.out.print("Enter the number of windows: ");
        roomWindows = input.nextInt();
```

Output:
Enter the length: 40
Enter the width: 75
Enter the height: 30
Enter the number of doors: 2
Enter the number of windows: 2
Paint required = 28.085714285714285 gallons

```java
        double totalSurfaceArea = 2 *
        (roomLength * roomWidth
          + roomLength * roomHeight
          + roomWidth * roomHeight);

        double floorArea = roomLength * roomWidth;

        int doorsArea = 20 * roomDoors;

        int windowsArea = 15 * roomWindows;

        double totalPaintArea = totalSurfaceArea - floorArea
                            - doorsArea - windowsArea;

        double requiredPaint = totalPaintArea / 350;

        System.out.println("Paint required = "
                + requiredPaint + " gallons");
    }
}
```

Exercises:

Exercise01. Write a Java program that reads a word as an input and converts that word as follows:

   a-) removing the first letter, and placing that letter at the end of the word

   b-) appending "oi" to the word. For example:

The original word: "happy"

The output: "appyhoi"

Exercise02. Write a Java program that prompts the user to enter two points (x1, y1) and (x2, y2) and displays their distance between them. The formula for computing the distance is

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Here is a sample run:

```
Enter x1 and y1: 1.5 -3.4  ↵Enter
Enter x2 and y2: 4 5  ↵Enter
The distance between the two points is 8.764131445842194
```

Homework:

Homework01. Write a Java program that reads in investment amount, annual interest rate, and number of years, and displays the future investment value using the following formula:

$$futureInvestmentValue =$$

$$investmentAmount \times (1 + monthlyInterestRate)^{numberOfYears*12}$$

For example, if you enter amount 1000, annual interest rate 3.25%, and number of years 1, the future investment value is 1032.98. 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
```

Lab 05
Chapter 03 Selections – Week 05

Objectives:

- To implement selection control using if statements.
- To implement selection control using switch statements.
- To write expressions using the conditional expression.

Examples:

Example01. Write a Java program that asks the user to enter three integers and then displays the largest one.

<pre>import java.util.Scanner;
 	
public class Ch03_Week05_E1{
  public static void main(String[] args)
    {
        int number1, number2, number3;
        int largestOne;
 	
        Scanner input = new Scanner(System.in);
 	
        System.out.print("Enter the first integer: ");
        number1 = input.nextInt();
 	
        System.out.print("Enter the second integer: ");
        number2 = input.nextInt();
 	
        System.out.print("Enter the third integer: ");
        number3 = input.nextInt();
 	
        if ((number1 > number2) && (number1 > number3))
        {
            largestOne = number1;
        }
        else if (number2 > number3)
        {
            largestOne = number2;
        }
        else
        {
            largestOne = number3;</pre>
	Output:
	Enter the first integer: 7
	Enter the second integer: 6
	Enter the third integer: 5
	The largest integer is 7
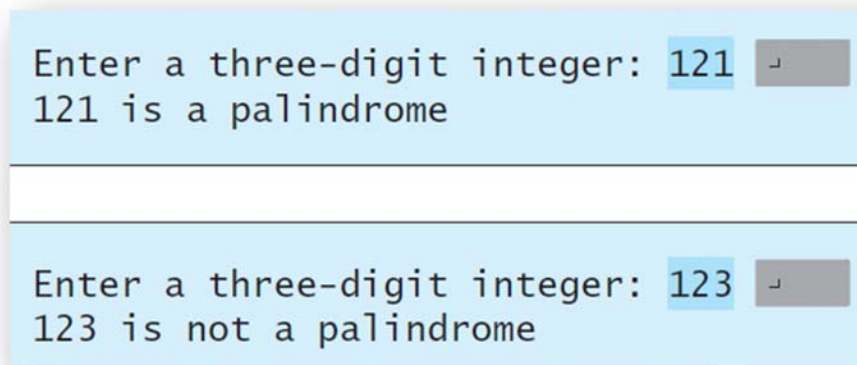
```
        }

        System.out.println("The largest integer is "
        + largestOne);
    }
}
```

Example02. Write a Java program that prompts the user to enter a grade. Your program should display the corresponding meaning of that grade using a switch statement.

``import java.util.Scanner;

public class Ch03_Week05_E2{
  public static void main(String[] args)
    {
        char userGrade;

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a grade: ");
        userGrade = input.next().charAt(0);

        switch (userGrade)
        {
        case 'A':
          System.out.println("\'A\' grade <==> Excellent");
          break;
        case 'B':
          System.out.println("\'B\' grade <==> Good");
          break;
        case 'C':
          System.out.println("\'C\' grade <==> Average");
          break;
        case 'D':
          System.out.println("\'D\' grade <==> Deficient");
          break;
        case 'F':
          System.out.println("\'F\' grade <==> Failing");
          break;
        default:
          System.out.println("Invalid grade");
        }
    }
}``` | Output:
Enter a grade: A
'A' grade <==>
Excellent |

Exercises:

Exercise01. Write a Java program that prompts the user to enter a three-digit integer and determines whether it is a palindrome number. A number is palindrome if it reads the same from right to left and from left to right. Here is a sample run of this program:

```
Enter a three-digit integer: 121 ↵
121 is a palindrome


Enter a three-digit integer: 123 ↵
123 is not a palindrome
```

Homework:

Homework01. An ISBN-10 (International Standard Book Number) consists of 10 digits: d1 d2 d3 d4 d5 d6 d7 d8 d9 d10. The last digit, d10, is a checksum, which is calculated from the other nine digits using the following formula:

(d1 * 1 + d2 * 2 + d3 * 3 + d4 * 4 + d5 * 5 +

d6 * 6 + d7 * 7 + d8 * 8 + d9 * 9) % 11

If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention. Write a Java 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
```

Challenging question (must be submitted in one week):

Please, write a program that does the following:

a-) Asking the customer about his/her name, then, storing the received name in a variable.

b-) Asking the customer about the number of phone calls that he/she made in the last 30 days, then, storing the received number of calls in a variable.

c-) Asking the customer if he/she has a VIP membership, then, storing the received answer in a variable of character data type. The customer can only enter character **Y** (for yes) or character **N** (for no).

d-) Using if statement to calculate the phone bill as follows:

d-1) if the number of calls is less than or equal to 100 calls and the customer has a VIP membership, then, $\textbf{phone bill} = \dfrac{(\textit{number of calls})\textbf{1.5}}{\textbf{0.5}}$

d-2) if the number of calls is less than or equal to 100 calls and the customer does not have a VIP membership, then, $\textbf{phone bill} = \dfrac{(\textit{number of calls})\textbf{2.2}}{\textbf{0.3}}$

d-3) if the number of calls exceeds 100 calls and the customer has a VIP membership, then, $\textbf{phone bill} = \dfrac{(\textit{number of calls})\textbf{2.5}}{\textbf{1.3}}$

d-4) if the number of calls exceeds 100 calls and the customer does not have a VIP membership, then, $\textbf{phone bill} = \dfrac{(\textit{number of calls})\textbf{2.7}}{\textbf{1.2}}$

e-) After calculating the phone bill, the following must be displayed on the screen:

Customer's name:	**?**	<--------- ? (received from the customer)
Number of calls:	**?**	<--------- ? (received from the customer)
VIP member:	**?**	<--------- could be **Y** or **N,** received from the customer
Phone bill=	**?**	<--------- ? the calculated bill

Lab 06
Chapter 05 Loops – Week 06

Objectives:

- To write loops using while statements.
- To write loops using do-while statements.
- To control a loop with a sentinel value.

Examples:

Example01. Write a Java program that prompts the user to enter a positive integer in order to print the multiplication table of that number.

	Output:

``java
import java.util.Scanner;
 	
public class Ch05_Week06_E1{
  public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int number;
 	
        System.out.print("Enter a positive integer: ");
        number = input.nextInt();
 	
System.out.println("The multiplication table of " + number);
 	
        for(int i=1; i<=10; i++)
        {
          System.out.println(number +" x " + i + " = "
          + (number * i));
        }
    }
}```
	The multiplication table of 9
	9 x 1 = 9
	9 x 2 = 18
	9 x 3 = 27
	9 x 4 = 36
	9 x 5 = 45
	9 x 6 = 54
	9 x 7 = 63
	9 x 8 = 72
	9 x 9 = 81
	9 x 10 = 90

Example02. Write a Java program that uses a do-while statement to find the factorial value of any positive integer received from the user, and the program will be terminated only if the user entered 999.

Code	Output
| ```java
import java.util.Scanner;

public class Ch05_Week06_E2{
  public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int userInput;
        int factorial;
        int i;

        do
        {
            System.out.print("Enter a positive integer: ");
            userInput = input.nextInt();

            if(userInput == 999)
            {
    System.out.println("The program will be terminated...\n"
                +"Good Bye...");
                System.exit(0);
            }

            factorial=1;
            i=1;
            while(i<=userInput)
            {
                factorial *= i;

                i++;
            }

            System.out.println("The factorial of  "
            + userInput
            + " = " + factorial);

        }while(userInput != 999);

    }
}
``` | Output:<br>Enter a positive integer:<br>5<br>The factorial of  5 = 120<br>Enter a positive integer:<br>6<br>The factorial of  6 = 720<br>Enter a positive integer:<br>999<br>The program will be terminated…<br>Good Bye… |

Exercises:

Exercise01. Write a Java program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input 0. Display the average as a floating-point number. Here is a sample run:

```
Enter an integer, the input ends if it is 0: 1 2 -1 3 0  ↵ Enter
The number of positives is 3
The number of negatives is 1
The total is 5.0
The average is 1.25
```

Homework:

Homework01. Write a Java program that converts kilograms (from 1 to 199) to pounds and converts pounds (from 20 to 515) to kilograms. The following two tables must be displayed side by side:

```
Kilograms       Pounds      |     Pounds        Kilograms
1                  2.2      |     20                9.09
3                  6.6      |     25               11.36
5                 11.0      |     30               13.64
---                ---      |     ---               ---
---                ---      |     ---               ---
197              433.4      |     510             231.82
199              437.8      |     515             234.09
```

Lab 07
Chapter 05 Loops – Week 07

Objectives:

- To write loops using for statements.
- To write nested loops.
- To implement program control with break and continue.

Examples:

Example01. Write a Java program that uses a for statement to sum the even integers from 5 to 35 and store the result in an integer variable called total.

| ```java
public class Ch05_Week07_E1{
  public static void main(String[] args)
  {
    int total=0;

    for(int i=5; i<=35; i++)
        total+=i;

    System.out.println("The total is "+total);
  }
}
``` | Output:<br>The total is 620 |

Example02. Write a Java program that uses nested for loops to display a specific pattern.

	Output:
| ```java
public class Ch05_Week07_E2{
  public static void main(String[] args)
  {
     int number=16;

    for(int i=5; i>=1; i--)
       {
           for(int j=1; j<=i; j++)
           {
               number=number - 1;
               System.out.printf("%-3d", number);
           }
           System.out.println();
       }
  }
}
``` | 15 14 13 12 11<br>10 9  8  7<br>6  5  4<br>3  2<br>1 |

Example03. Write a Java program that uses a continue statement to find the sum of odd integers only, and uses a break statement to end the loop if the sum is greater than 500.

	Output:
| ```java
public class Ch05_Week07_E3{
  public static void main(String[] args)
  {
     int sum=0;

     for(int i=1; i<=1000; i++)
     {
        if(i % 2 == 0)
          continue;

        sum += i;

       if(sum > 500)
         break;
    }

    System.out.println("Sum = "+ sum);
  }
}
``` | Sum = 529 |

Exercises:

Exercise01. Write a Java program that prompts the user to enter the number of students and each student's name and score, and finally displays the student with the highest score and the student with the second-highest score.

Homework:

Homework01. Write a Java program that prompts the user to enter the year and first day of the year and displays the calendar table for the year on the console. For example, if the user entered the year 2013, and 2 for Tuesday, January 1, 2013, your program should display the calendar for each month in the year, as follows:

		January 2013				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Challenging question (must be submitted in one week):

Write a Java program that uses nested for loops to display the following figure:

```
* * * * * *  o
*        */|\
*        */ \
*        * * * * *  o
*                */|\
*                */ \
*                * * * * *  o
*                        */|\
*                        */ \
*                        * * * * *  o
*                                */|\
*                                */ \
*                                * * * * *  o
*                                        */|\
*                                        */ \
*                                        * * * * *  o
*                                                */|\
*                                                */ \
* * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Lab 08
Chapter 06 Methods – Week 08

Objectives*:*

- To define and call `void` methods with formal parameters and with actual parameters (i.e., arguments).
- To define and call methods with a return value.

Examples:

Example01. Write Java program to define and call void `myMethod()` function. `myMethod()` function prints a message "I just got executed!":

	Output:
| ```java
public class Ch08_Week08_E1 {
  static void myMethod() {
    System.out.println("I just got called
                       and executed!");
  }
  public static void main(String[] args) {
    myMethod();
  }
}
``` | I just got called and executed! |

Example02. Write Java program to define void myMethod() function with a single formal parameter 'fname', where myMethod(String fname) function print a string. Call myMethod() function with a single actual parameter "*Muhammad*":

	Output:
| ```java
public class Ch06_Week08_E2 {
  static void myMethod(String fname) {
    System.out.println(fname + " Al-Ali");
  }

  public static void main(String[] args) {
    myMethod("Muhammad");
  }
}
``` | Muhammad Al-Ali |

Example03. Write Java program to define and call myMethod(int firstNumber, int secondNumber) function with a return value, where myMethod(int firstNumber, int secondNumber) function returns the sum of firstNumber and secondNumber parameters:

| ```java
public class Ch06_Week08_E3{
  static int myMethod(int firstNumber,
                     int secondNumber) {
    return firstNumber + secondNumber;
  }

  public static void main(String[] args) {
    System.out.println(myMethod(4, 6));
  }
}
``` | Output:<br><br>10 |

Exercises:

Exercise01. Write a class that contains the following two methods:

- public static double footToMeter(double foot)   /** Convert from feet to meters */
- public static double meterToFoot(double meter)   /** Convert from meters to feet */

Where the formula for the conversion is 1 meter = 0.305 * foot, and 1 foot = 3.279 * meter. Consider the following values feets (1.0, 2.0, 3.0) and meters (20.0, 21.0, 22.0):
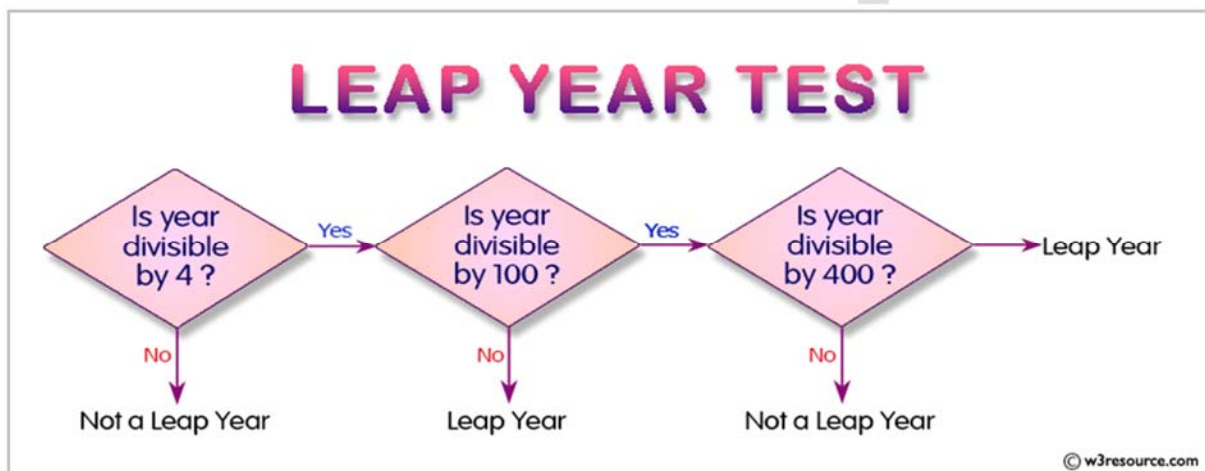
Homework:

Homework01. Write a method that returns the number of days in a year using the following header:

<p style="text-align:center;">*public static int numberOfDaysInAYear(int year)*</p>

Write a test program that displays the number of days in year from 2000 to 2010. The following figure explain whether a year is a leap or not. Use to show the number of the days in the year.

Lab 09
Chapter 06 Methods – Week 09

Objectives:

- To pass arguments by value.
- To use method overloading.
- To determine the scope of variables.

Examples:

Example01. Write Java program to show the value of x variable before and after passing a value to *increment(int n)* function:

``java
public class Ch06_Week09_Example01 {
 	
 public static void increment(int n) {
  n++;
  System.out.println("n inside the method is " + n);
}
 	
 public static void main(String[] args) {
  int x = 1;
  System.out.println("Before the call, x is " + x);
  increment(x);
  System.out.println("After the call, x is " + x);
 }
}
``
	Output:
	
	Before the call, x is 1
	n inside the method is 2
	After the call, x is 1

Example02. Write Java program to create two functions named *max*. The first max(int num1, int num2) finds the maximum integer between two integer number, and the second max(double num1, double num2) finds the maximum double between two double numbers:

| ```java
public class Ch06_Week09_Example02 {

public static void main(String[] args) {
 int num1 =0;
 int num2 =1;

 System.out.println("The max of 3 and 4 is "+ max(3, 4));
 System.out.println("The max of 9.45 and 2.9 is " +
                  max(9.45, 2.9));
 }

public static int max(int num1, int num2) {
 if (num1 > num2) return num1;
 else
 return num2;
 }

public static double max(double num1, double num2) {
 if (num1 > num2) return num1;
 else
  return num2;
 }
}
``` | Output:<br><br>The max of 3 and 4 is 4<br>The max of 9.45 and 2.9 is 9.45 |

Exercises:

Exercise01. Write a Java program to create a method CountVowelsInAWord(string word) that count all vowels in a string. For example, the word "taibah university" has 7 vowels.

Note: These letters are vowels in English: a, e, i, o, u.

Homework:

Homework01. Write Java methods to calculate the area of a triangle. Define and call CalculateAreaTriangle(double firstSide, double secondSide, double thirdSide) for the following three triangles: first triangle(10, 15, 20), second triangle(4, 6, 8), and third triangle(15, 10, 5). Were all triangles valid or not ?

Note: the following figure shows how to calculate the area of triangle.

Lab 10
Chapter 06 Methods – Week 10

Objectives:

- To determine the scope of variables.
- To solve mathematical problems by using the methods in the Math class

Examples:

Example01. Write Java program with the same method twice where it takes a single argument. The program prints the value of x squared to the screen:

      a. What is the difference between x and y variables in terms of scope ?

| ```java
public class Ch06_Week10_Example01 {

    public static void printSquare(double y)
    {
        System.out.println(y*y);
        return;
    }

    public static void main(String[] argv)
    {
      double x = 5.0;
      printSquare(3.0);
      printSquare(x);
    }
}
``` | Output:<br><br>9.0<br>25.0 |
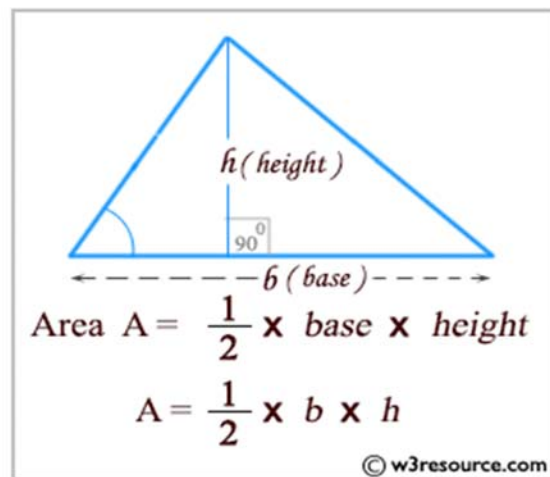
Example02. Write Java program to call suitable functions from the Math class solve the following mathematical questions:

- Round a floating-point value up of `7.343` to the nearest integer value.
- Find the value of 2 raised to the power of 8.
- Find the logarithm of 1.
- Find the cosine value of 45° in radians.

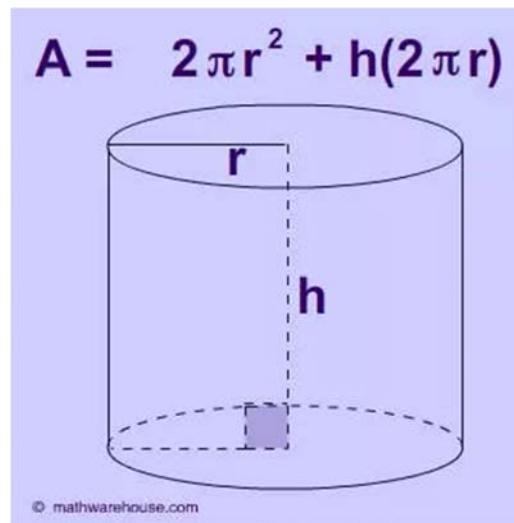|	Output:

```java`import java.lang.Math;`
 	
`public class Ch06_Week10_Example02 {`
 	
`public static void main(String[] args) {`
 	
`double ceil = Math.ceil(7.343);`
`System.out.println("Ceiling of 7.343 is " + ceil);`
 	
`double pow = Math.pow(2,8);`
`System.out.println("2 to the power of 8 is " + pow);`
 	
`double log1  = Math.log(1);`
`System.out.println("Logarithm of 1 is " + log1);`
 	
`double cos = Math.cos(45);`
`System.out.println("The cosine value of 45 is " + cos);`
`  }`
`}`
``
	Ceiling of 7.343 is 8.0
	2 to the power of 8 is 256.0
	Logarithm of 1 is 0.0
	The cosine value of 45 is 0.525

Exercises:
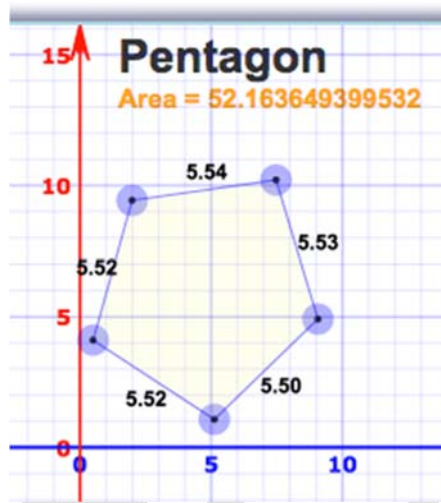
Exercise01. Write a Java program to define and call CalculateAreaCylinder(double height, double radius) function to calculate the area of three cylinders using Math class with the following specifications: first cylinder (7,6), second cylinder (4.3 ,2.5), and third cylinder (-4.5,6.9).

Note the area of cylinder is calculated using the following formulae:

$$A = 2\pi r^2 + h(2\pi r)$$

Homework:

Homework01. Write a Java program to define CalculateAreaPentagon(double numberOfSides, double sideLength) function to calculate the area of two pentagons with the following specification: first pentagon(5, 5.5), and second pentagon(8, 2.3) using the Math class. Allow the user to enter the number of sides for each pentagon, enter the length of the side, and computes the area of the pentagon, as shown in the following figure:

Note: The formula for computing the area of a pentagon is Area =
$\frac{numberOfSides \times sideLength^2}{4 \times tan(\frac{\pi}{numberOfSides})}$. *Round up two digits after the decimal point.*

Challenging question (must be submitted in one week):

Use the methods in RandomCharacter.java to print 100 uppercase letters and then 100 single digits, printing ten per line.

```
// RandomCharacter.java

public class RandomCharacter {
    /** Denerate a random character between ch1 and ch2 */
    public static char getRandomCharacter(char ch1, char ch2) {
        return (char)(ch1 + Math.random() * (ch2 - ch1 + 1));
    }
    /** Generate a random lowercase letter */
    public static char getRandomLowerCaseLetter() {
        return getRandomCharacter('a', 'z');
    }
    /** Generate a random uppercase letter */
    public static char getRandomUpperCaseLetter() {
        return getRandomCharacter('A', 'Z');
    }
    /** Generate a random digit character */
    public static char getRandomDigitCharacter() {
        return getRandomCharacter('0', '9');
    }
    /** Generate a random character */
    public static char getRandomCharacter() {
        return getRandomCharacter('\u0000', '\uFFFF');
    }
}
```

Lab 11
Chapter 07 Single-dimensional Arrays – Week 11

Objectives*:*

- To create single-dimensional arrays and access array elements using indexes.
- To initialize an array using an array initializer.
- To program common array operations (displaying arrays, summing all elements)
- To invoke methods with array arguments and return values.

Examples:

Example01. Write Java program to create two 1-dimensional arrays, myFirstArray and mySecondArray, with the following specifications:

- Declare myFirstArray variable with int type.
- Initialize mySecondArray variable with double type and size of 5 using Array Initializer.
- Assign five elements of double type with 5.6, 4.5, 3.3, 13.2, and 4.0 values, and assigns its reference to mySecondArray.
- Add 20 on the first index of mySecondArray.
- Print the first index, middle index, and the last index of mySecondArray.

| ```java
public class Ch07_Week11_Example01 {

public static void main(String[] args) {

int[] myFirstArray;
double[] mySecondArray = new double[5];

mySecondArray[0] = 5.6;
mySecondArray[1] = 4.5;
mySecondArray[2] = 3.3;
mySecondArray[3] = 13.2;
mySecondArray[4] = 4.0;


mySecondArray[0] = mySecondArray[0] + 20;


System.out.println("The first index is " + mySecondArray[0]);
System.out.println("The middle index is " + mySecondArray[2]);
System.out.println("The last index is " + mySecondArray[4]);
     }     }
``` | Output:<br><br>The first index is 25.6<br><br>The middle index is 3.3<br><br>The last index is 4.0 |
|---|---|

Example02. Write Java program to process 1-dimensional array, myThirdArray and myFourthArray with the following specifications:

- Initialize myThirdArray with String type with size 7.
- Allow the user to input values using for loop.
- Print the myThirdArray using a for loop.

	Output:
| ```java
import java.util.Scanner;

public class Ch07_Week11_Example02 {

public static void main(String[] args) {

String[] myThirdArray = new String[7];
java.util.Scanner input = new java.util.Scanner(System.in);

System.out.println("Enter " + myThirdArray.length + " values: ");

for (int i = 0; i < myThirdArray.length; i++) {
   myThirdArray[i] = input.next();
}

for (int j = 0; j < myThirdArray.length; j++) {
   System.out.print(myThirdArray[j]);
      }
   }
}
``` | Enter 7 values:<br>T<br>I<br>A<br>B<br>A<br>H<br>U<br>TIABAHU |

Example03. Write Java program to process 1-dimensional array, myFourthArray with the following specifications:

- Initialize myFourthArray with int type with size 7.
- Define a function, FindSumElementsArray(int arrayName) that allows the user to input values using for loop and calculate and total of its elements.
- Use a variable named total to store the sum of the myFourthArray. Initially total is 0. Add each element in the array to total using the loop.

	Output:
| ```java
public class Ch07_Week11_Example03 {

public static int FindSumElementsArray(int[] arrayName){
 int total = 0;
 for (int n = 0; n < arrayName.length; n++) {
 total += arrayName[n];
 //System.out.println("total is " + total);
 }
 return total;
}

public static void main(String[] args) {

 int[] myFourthArray = new int[7];

 java.util.Scanner input2 = new java.util.Scanner(System.in);
 System.out.println("Enter " + myFourthArray.length + " values:
");

  for (int k = 0; k < myFourthArray.length; k++) {
    myFourthArray[k] = input2.nextInt();
  }

  int total = FindSumElementsArray(myFourthArray);
  System.out.println("total is " + total);
  }
}
``` | Enter 7 values:<br>1<br>2<br>3<br>4<br>5<br>6<br>7<br>total is 28 |

Exercises:

Exercise01. Write a Java program to create a method that finds the smallest element in an array of double values using the following header:

```
public static double min(double[] array)
```

Write a test program that prompts the user to enter ten numbers, invokes this method to return the minimum value, and displays the minimum value.

Note: you may use For-each Loop.

Homework:

Homework01. 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 prompts the user to enter ten double values, invokes this method, and displays the average value.

Note: you may use For-each Loop.

Challenging question (must be submitted in one week):

Write a program that picks four cards from a deck of 52 cards and computes their sum. An Ace, King, Queen, and Jack represent 1, 13, 12, and 11, respectively. Your program should display the number of picks that yields the sum of 24.

Lab 12
Chapter 08 Multidimensional Arrays – Week 12

Objectives:

- To declare two-dimensional arrays, and access array elements in a two-dimensional array using row and column indexes.
- To program common operations for two-dimensional arrays (displaying arrays, summing all elements).
- To create and use multidimensional arrays.

Examples:

Example01. Write a Java program to create two multi-dimensional arrays, myFirstMatrix, and mySecondMatrix with the following specifications:

- Initialize myFirstMatrix variable with int type and size of 5x5 or 5-by-5 (i.e., five rows, five columns).
- Print the value of first index, and the value of last index of myFirstMatrix.
- Initialize mySecondMatrix variable with double type and size of 4-by-3 (i.e., four rows, 3 columns) using Array Initializer.
- Assign the following elements of double type with (1.0, 2.0, 3.0), (4.0, 5.0, 6.0), (7.0, 8.0, 9.0), (10.0, 11.0, 12.0), and assigns its reference to mySecondMatrix.
- Print the value of first index, and the value of last index of mySecondMatrix.

```public class Ch08_Week12_Example01 {```	Output:
```public static void main(String[] args) {```	The value of first index of myFirstMatrix is 0
```int[][] myFirstMatrix;```	
```int firstMatrixRowNum = 5;```	
```int firstMatrixColumnNum = 5;```	The value of last index of myFirstMatrix is 0
```myFirstMatrix= new int[firstMatrixRowNum][firstMatrixColumnNum];```	
```System.out.println("The value of first index of myFirstMatrix is``` ``` " + myFirstMatrix[0][0]);```	The value of the first index of mySecondMatrix is 1.0
```System.out.println("The value of last index of myFirstMatrix is``` ``` " + myFirstMatrix[4][4]);```	
```double[][] mySecondMatrix = {{1.0, 2.0, 3.0},``` ``` {4, 5, 6},```	

`{7.0, 8, 9.0},`
                                  `{10, 11.0, 12}};`
 	
`System.out.println("The value of the first index of`
`mySecondMatrix is " + mySecondMatrix[0][0]);`
 	
`System.out.println("The value of the last index of`
`mySecondMatrix is " + mySecondMatrix[3][2]);`
   `}`
`}`
	The value of the last index of mySecondMatrix is 12.0

Example02. Write a Java program to create a two-dimensional array, myThirdMatrix, with the following specifications:

- Initialize myThirdMatrix with int type and size of 2-by-4.
- Allow the user to input values using for loop. Use `length()` function to get the array length.
- Print the myThirdMatrix using a for loop.

`import java.util.Scanner;`
 	
`public class Ch08_Week12_Example02 {`
 	
`public static void main(String[] args) {`
 	
`java.util.Scanner input = new Scanner(System.in);`
 	
`int[][] myThirdMatrix = new int[2][4];`
 	
`System.out.println("Enter " + myThirdMatrix.length + " rows and "`
                   `+ myThirdMatrix[0].length + " columns: ");`
`for (int row = 0; row < myThirdMatrix.length; row++) {`
    `for (int column = 0; column < myThirdMatrix[row].length;`
       `column++) {`
        `myThirdMatrix[row][column] = input.nextInt();`
           `}`
        `}`
`for (int row = 0; row < myThirdMatrix.length; row++) {`
    `for (int column = 0; column < myThirdMatrix[row].length;`
       `column++) {`
       `System.out.print(myThirdMatrix[row][column] + " "); }`
          `System.out.println();`
      `}`   `}`   `}`
	Output:
	
	Enter 2 rows and 4 columns:
	2
	4
	6
	8
	1
	3
	5
	7
	
	2 4 6 8
	1 3 5 7

Example03. Write a Java program to create a three-dimensional array, myFourthMatrix, with the following specifications:

- Initialize myFourthMatrix with int type and size of 2x3x4 (i.e., 2 rows, 3 columns, 4 heights).
- Use a for loop to fill the myFourthMatrix with random int values between 0 and 10.
- Print the myFourthMatrix using a for loop.

| ```java
public class Ch08_Week12_Example03 {

public static void main(String[] args) {

int[][][] myFourthMatrix  = new int[2][3][4];

for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 3; j++) {
        for (int k = 0; k < 4; k++) {
        myFourthMatrix [i][j][k] = (int)(Math.random() * 10);
                }
            }
        }

for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 3; j++) {
        for (int k = 0; k < 4; k++) {
        System.out.print(myFourthMatrix [i][j][k] + " ");
                }
            System.out.println();
            }
        }
    }
}
``` | Output:<br><br>2 3 9 5<br>8 7 9 4<br>9 5 6 8<br>8 3 8 5<br>5 2 6 0<br>0 1 9 6 |
|---|---|

Exercises:

Exercise01. Write a Java program to create a two-dimensional array, myFifthMatrix, with the following specifications:

- Initialize myFifthMatrix with int type and with size [3][3].
- Use a for loop to fill the myFifthMatrix with random values between 0 and 99.
- Print the myFifthMatrix using a for loop.
- For each column, use a variable named total to store its sum. Add each element in the column to total using a for loop.

Homework:

Homework01. Write a Java program to create three metrics with int type and size of 2x2. Allow the user to fill two metrics with the following values: {(1,2),(3,4)}, and {(9,8),(7,6)}. The third matrix will be filled with the sum of two metrics. Write a method to add the two matrices. The header of the method is as follows:

    public static double[][] addMatrix(double[][] a, double[][] b)

*Note: In order to be added, the two matrices must have the same dimensions and the same or compatible types of elements. Let c be the resulting matrix. Each element c_{ij} is $a_{ij} + b_{ij}$. For example, for two 2 * 2 matrices a and b, c is*

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{pmatrix}$$

Challenging question (must be submitted in two weeks):

In a game of tic-tac-toe, two players take turns marking an available cell in a 3x3 grid with their respective tokens (either X or O). When one player has placed three tokens in a horizontal, vertical, or diagonal row on the grid, the game is over, and that player has won. A draw (no winner) occurs when all the cells on the grid have been filled with tokens and neither player has achieved a win. Create a Java program for playing tic-tac-toe using single- and multi-dimensional arrays . The program prompts two players to enter an X token and O token alternately. Whenever a token is entered, the program redisplays the board on the console and determines the status of the game (win, draw, or continue). Here is a sample run:

```
            -------------
            |   |   |   |
            -------------
            |   |   |   |
            -------------
            |   |   |   |
            -------------
    Enter a row(0, 1, or 2) for player  X : 1
    Enter a column(0, 1, or 2) for player  X : 1

            -------------
            |   |   |   |
            -------------
            |   | X |   |
            -------------
            |   |   |   |
            -------------
    Enter a row(0, 1, or 2) for player  O : 0
    Enter a column(0, 1, or 2) for player  O : 0

            -------------
            | O |   |   |
            -------------
            |   | X |   |
            -------------
            |   |   |   |
            -------------
    Enter a row(0, 1, or 2) for player  X : 2
    Enter a column(0, 1, or 2) for player  X : 0

            -------------
            | O |   |   |
            -------------
            |   | X |   |
            -------------
```

```
                        | X |   |   |
                        -------------
        Enter a row(0, 1, or 2) for player  O : 2
        Enter a column(0, 1, or 2) for player  O : 2


                        -------------
                        | O |   |   |
                        -------------
                        |   | X |   |
                        -------------
                        | X |   | O |
                        -------------
        Enter a row(0, 1, or 2) for player  X : 0
        Enter a column(0, 1, or 2) for player  X : 2


                        -------------
                        | O |   | X |
                        -------------
                        |   | X |   |
                        -------------
                        | X |   | O |
                        -------------
                        X player won
```

Note: Use the following test program:
public static void main(String[] args) {

String [][] board = getBoard();   // Create an empty board

String[] tokens = {" X "," O "};   // Create two players token

int result;   // game status result

do {   // Repeat while result is continue

print(board);   // Display board

int[] cell = getCell(board, tokens[0]); // Get available cell to mark

placeToken(board, cell, tokens[0]); // Mark available cell
             with player's token

result = gameStatus(board, tokens[0]); // Determine game status

if (result == 2) {   // If status is continue make next
            player take turn

swap(tokens);
}

} while(result == 2);

print(board);   // Display board

```
if (result == 0)                          // Display game results

System.out.println(tokens[0] + "player won");

else

System.out.println("Players draw");

}
```

Lab 13
Chapter 09 Objects and Classes – Week 13

Objectives:

- To demonstrate how to define classes and create objects.
- To access objects via object reference variables.
- To create objects using constructors.

Examples:

Example01. Write Java program to create the Employee class. The Employee class has three variables – name with string type, age with int type, and salary with double type, and PrintMessage() function. PrintMessage() shows a message whenever an instance is created. Write a test program that declare and creates two objects or instances of the Employee class, Ali and Saeed.

Code	Output:

``java
// Employee.java
public class Employee {
 	
    String name;
    int age;
    double salary;
 	
    public void PrintMessage() {
       System.out.println("An Employee created.");
    }
}```	

Code	Output:

``java
// Ch09_Week13_Example01.java
 	
public class Ch09_Week13_Example01 {
 	
public static void main(String []args) {
Employee Ali = new Employee();
Ali.PrintMessage();
 	
Employee Saeed = new Employee();
Saeed.PrintMessage();
      }
}```
	An Employee created.
	An Employee created.

Example02. The EmployeeWithConstructors class is a modified version of the Employee class from Example01. Using the EmployeeWithConstructors class, add four constructors: Employee(), EmployeeName(String employeeName), EmployeeAge(int employeeAge), and EmployeeSalary(double employeeSalary). Employee() is the default constructor, and the EmployeeName() assigns a string value, EmployeeAge() assigns an int value, and EmployeeSalary() assigns a double value. Also, create *PrintEmployeeInfo*() function to print the created instance information. Write Java program to create two instances from the EmployeeWithConstructors class : Maha, and Arwa. Arwa has the following information: (Name: Arwa, Age: 29, Salary: 5500.59).

	Output:
| ```java
// EmployeeWithConstructors.java
public class EmployeeWithConstructors {

    String name;
    int age;
    double salary;

     public EmployeeWithConstructors() {
        System.out.println("Default constructor invoked.");
    }

    public void EmployeeName(String employeeName) {
        this.name = employeeName;
    }

    public void EmployeeAge(int employeeAge) {
        this.age = employeeAge;
    }

    public void EmployeeSalary(double employeeSalary) {
        this.salary = employeeSalary;
    }

    public void PrintEmployeeInfo() {
        System.out.println("Name: "+ name );
        System.out.println("Age: " + age );
        System.out.println("Salary: " + salary );
    }
}
``` | |

	Output:
| ```java
// Ch09_Week13_Example02

public class Ch09_Week13_Example02 {

public static void main(String []args) {

EmployeeWithConstructors Maha = new EmployeeWithConstructors();

EmployeeWithConstructors Arwa = new EmployeeWithConstructors();
Arwa.EmployeeName("Arwa");
Arwa.EmployeeAge(29);
Arwa.EmployeeSalary(5500.59);

Arwa.PrintEmployeeInfo();
      }
}
``` | Default constructor invoked.<br><br>Default constructor invoked.<br><br>Name: Arwa<br><br>Age: 29<br><br>Salary: 5500.59 |
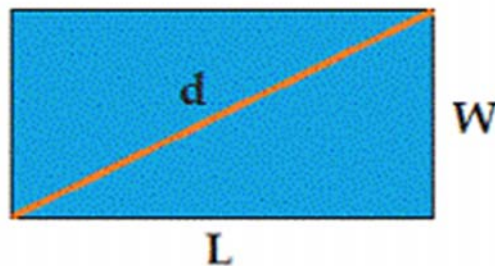
Exercises:

Exercise01. Design a class named Rectangle to represent a rectangle. The class contains:

- Two double data fields named width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.
- A no-arg constructor that creates a default rectangle.
- A constructor that creates a rectangle with the specified width and height.
- A method named getArea() that returns the area of this rectangle.
- A method named getPerimeter() that returns the perimeter.

Write a test program that creates two Rectangle objects—one with width 4 and height 40 and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this order.

Note: The area and perimeter of rectangle will be calculated as the following:

$$Area = length * width$$

$$Perimeter = 2L + 2W$$

Homework:

Homework01. Design a class named Account that contains:
- A private int data field named id for the account (default 0).
- A private double data field named balance for the account (default 0).
- A private Date data field named dateCreated that stores the date when the account was created.
- A no-arg constructor that creates a default account.
- A constructor that creates an account with the specified id and initial balance.
- A method named withdraw that withdraws a specified amount from the account.
- A method named deposit that deposits a specified amount to the account.

Write a test program that creates an Account object with an account ID of 1122, a balance of $20,000. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the balance, and the date when this account was created.