

Home Assignment Two

Programming Fundamentals (Java)

Registration # Fa20-Bse-094

Muhammad Ruslan Babar

Submitted to: Sir Azfar Shakeel Khan.

**Question 1:** Write a program (using nested if statements) 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.

**Source Code:**

package Russi7kd;

import java.util.\*;

public class Password\_Validation\_Nested\_if {

public static void main(String[] args) {

Scanner input = new Scanner (System.in);

System.out.println("\*\*\*\* Password Validator \*\*\*\*");

System.out.print("Enter password for validation:");

String password = input.next();

// Method Caller

if (isValid(password))

System.out.println("Valid Password");

else

System.out.println("Invalid Password");

}

// isValid method For password Validation

public static boolean isValid(String passCode){

boolean letters\_Digits = true;

int digit\_counter = 0;

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

if (!(Character.isLetter(passCode.charAt(i)) || Character.isDigit(passCode.charAt(i)))){

letters\_Digits = false;

break;

}

if (Character.isDigit(passCode.charAt(i)))

digit\_counter ++;

}

// Length Finding

if (passCode.length() >=8)

// Only Containing Letters And Digits

if (letters\_Digits)

// Atleast Digits

if (digit\_counter >=2)

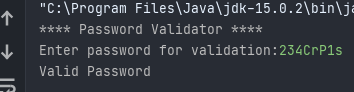
return true;

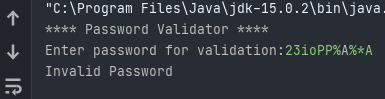
return false;

}

}

**Output shot:**





**Question 2:**

Write a method that sums all the numbers in the major diagonal in an n\*n matrix of double values using the following header:

* public static double sumMajorDiagonal(double[][] m)

Write a test program that reads a 4-by-4 matrix and displays the sum of all its elements on the major diagonal.

**Source Code:**

package Russi7kd;

import java.util.\*;

public class Major\_Diagonal {

public static void main(String[] args) {

double [][] matrix = new double[4][4];

System.out.println("Enter a 4-by-4 matrix row by row: ");

getInput(matrix); //Completing 2D list

double results = sumMajorDiagonal(matrix); //Calculating sumMajorDiagonal

System.out.println("Sum of elements in major diagonal is: "+results);

}

//Method for getting input

public static void getInput(double[][] matrix){

Scanner input = new Scanner (System.in);

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

System.out.print("\nComplete row "+(i+1)+" ");

for (int j = 0; j < matrix[i].length; j++) {

matrix[i][j] = input.nextDouble();

}

}

}

public static double sumMajorDiagonal(double[][] matrix){

double sum = 0.0;

//Calculating sum of major diagonal

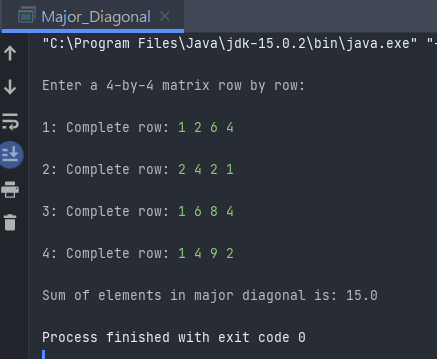
for (int i = 0; i < matrix.length; i++)

sum += matrix[i][i];

return sum;

}

}

**OUTPUT:**

**Question 3:**

(Palindrome integer) Write the methods with the following headers

 // Return the reversal of an integer, i.e., reverse(456) returns 654

public static int reverse(int number)

// Return true if number is a palindrome

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.

**Source Code:**

package russi;

import java.util.\*;

public class Palendrome\_Integer {

public static void main(String[] args) {

Scanner input = new Scanner (System.in);

System.out.print("Enter a number to check Palindrome: ");

int number = input.nextInt();

//Method Caller

boolean isPal = isPalindrome(number);

//Checking returned Value

if(isPal)

System.out.println("The number: "+number+" is Palindrome");

else

System.out.println("The number: "+number+" is not Palindrome");

}

//Palindrome Method

public static boolean isPalindrome(int number){

if (number == reverse(number)) {

return true;

}

return false;

}

//reversing method

public static int reverse(int number){

String stringNum = "" + number;

String reverseNumber = "";

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

reverseNumber = stringNum.charAt(i) + reverseNumber;

return (Integer.parseInt(reverseNumber));

}

}

**Output shot:**

