|  |
| --- |
| **Lab03: Methods and Recursion** |

Designing and implementing Java programs that deal with:

|  |
| --- |
| 1. Static Methods 2. Recursion |

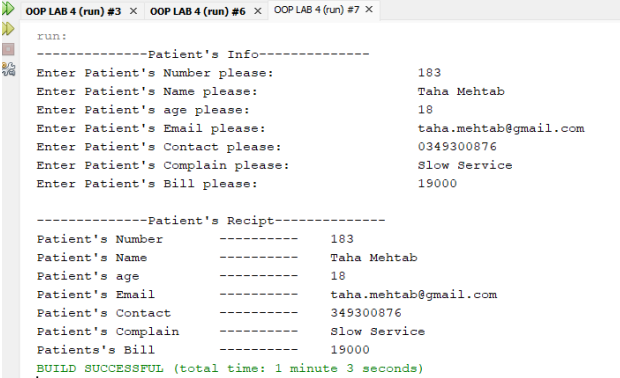
|  |
| --- |
| **Exercises** |

Exercise 1 (PatientInfo.java)

Consider you are a receptionist at hospital and whenever the patient comes you are to take his following info pNumber, pName, pAge, pEmail, pContact, pComplain and pBill then print the receipt for customer so method responsible for taking customer’s info is called as Take\_Patient\_data() and method responsible print receipt is called as Print\_Receipt()

Hint: Create global variable that is outside of the main method and use them in both methods for taking and printing customer’s details.

NOTE: These functions must not be static.



SOURCE CODE:

package Patientinfo;

import java.util.Scanner;

public class PatientInfo {

// Global variables to store patient details

String pNumber;

String pName;

int pAge;

String pEmail;

String pContact;

String pComplain;

double pBill;

// Method to take patient data

public void Take\_Patient\_data() {

Scanner scanner = new Scanner(System.in);

System.out.println("-------------Patient's Info-------------");

System.out.print("Enter Patient's Number please: ");

pNumber = scanner.nextLine();

System.out.print("Enter Patient's Name please: ");

pName = scanner.nextLine();

System.out.print("Enter Patient's age please: ");

pAge = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter Patient's Email please: ");

pEmail = scanner.nextLine();

System.out.print("Enter Patient's Contact please: ");

pContact = scanner.nextLine();

System.out.print("Enter Patient's Complain please: ");

pComplain = scanner.nextLine();

System.out.print("Enter Patient's Bill please: ");

pBill = scanner.nextDouble();

}

// Method to print patient receipt without using printf

public void Print\_Receipt() {

System.out.println("\n-------------Patient's Receipt-------------");

System.out.println("Patient's Number: " + pNumber);

System.out.println("Patient's Name: " + pName);

System.out.println("Patient's age: " + pAge);

System.out.println("Patient's Email: " + pEmail);

System.out.println("Patient's Contact: " + pContact);

System.out.println("Patient's Complain: " + pComplain);

System.out.println("Patient's Bill: " + pBill);

System.out.println("------------------------------------------");

}

// Main method

public static void main(String[] args) {

// Create an object of the class

PatientInfo receptionist = new PatientInfo();

// Take patient data

receptionist.Take\_Patient\_data();

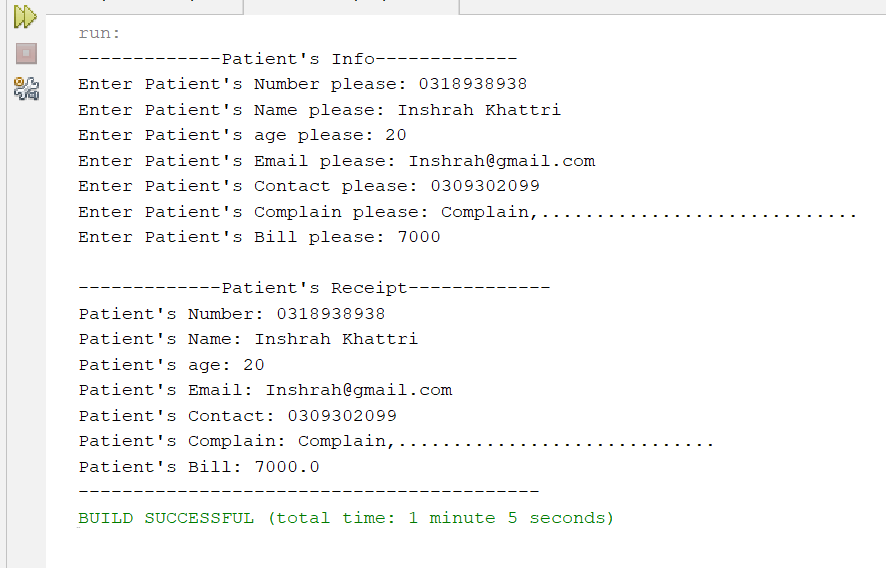
// Print receipt

receptionist.Print\_Receipt();

}

}

OUTPUT:



Exercise 2 (Sum.java)

Write the following 2 static methods:

1. public static int ComputeOddSum(int input)

(The method ComputeOddSum find the sum of all odd numbers less than input. *Should be recursive function.*

1. public static int ComputeEvenSum(int input)

(The method ComputeEvenSum find the sum of all even numbers less than input)

Now, test these 2 methods by prompting the user to input a number each time until a negative number is entered.

SOURCE CODE:

package sum;

import java.util.Scanner;

public class Sum {

public class SumCalculator {

public static int ComputeOddSum(int input) {

if (input <= 0) {

return 0;

}

if (input % 2 != 0) {

return input + ComputeOddSum(input - 2);

} else {

return ComputeOddSum(input - 1);

}

}

public static int ComputeEvenSum(int input) {

int sum = 0;

for (int i = 2; i < input; i += 2) {

sum += i;

}

return sum;

}

public static void main(String[] args) {

try (Scanner scanner = new Scanner(System.in)) {

while (true) {

System.out.print("Enter a number (negative number to quit): ");

int input = scanner.nextInt();

if (input < 0) {

System.out.println("Exiting...");

break;

}

int oddSum = ComputeOddSum(input);

int evenSum = ComputeEvenSum(input);

System.out.println("Sum of odd numbers less than " + input + " is: " + oddSum);

System.out.println("Sum of even numbers less than " + input + " is: " + evenSum);

}

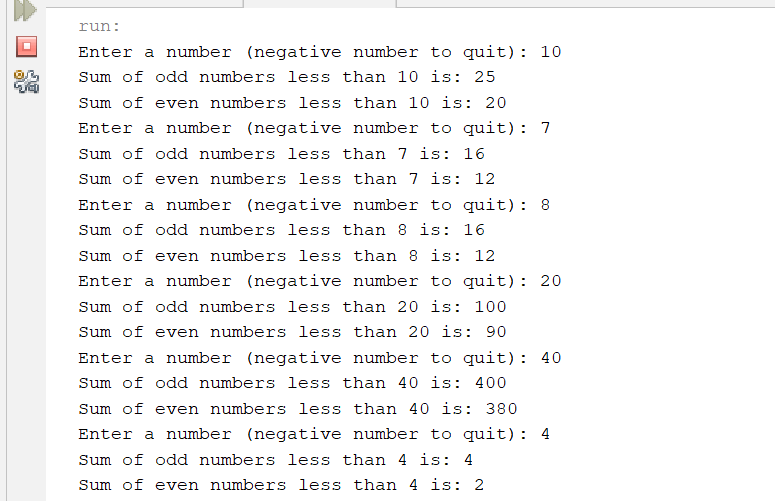
}

}

}

}

OUTPUT:



Exercise 3 (MatrixTest.java)

Create a Matrix named as Mat\_1 of size 3x3 and ask user to insert values take another matrix named as Mat\_2 of size 3x3 again and then implement following equations

(Mat\_1\*³) + (Mat\_2)²

(Mat\_2 -3) ²

(Mat\_2\*⁵) – (Mat\_1-2)

Source Code:

package matrixtest;

import java.util.Scanner;

public class MatrixTest {

public static void main(String[] args) {

// Declare two 3x3 matrices

try (Scanner scanner = new Scanner(System.in)) {

// Declare two 3x3 matrices

int[][] Mat\_1 = new int[3][3];

int[][] Mat\_2 = new int[3][3];

// Input values for Mat\_1

System.out.println("Enter values for Mat\_1 (3x3):");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

Mat\_1[i][j] = scanner.nextInt();

}

}

// Input values for Mat\_2

System.out.println("Enter values for Mat\_2 (3x3):");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

Mat\_2[i][j] = scanner.nextInt();

}

}

// Equation 1: (Mat\_1^3) + (Mat\_2^2)

System.out.println("\nResult of (Mat\_1^3) + (Mat\_2^2):");

int[][] result1 = new int[3][3];

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

result1[i][j] = (int) Math.pow(Mat\_1[i][j], 3) + (int) Math.pow(Mat\_2[i][j], 2);

System.out.print(result1[i][j] + "\t");

}

System.out.println();

}

// Equation 2: (Mat\_2 - 3)^2

System.out.println("\nResult of (Mat\_2 - 3)^2:");

int[][] result2 = new int[3][3];

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

result2[i][j] = (int) Math.pow((Mat\_2[i][j] - 3), 2);

System.out.print(result2[i][j] + "\t");

}

System.out.println();

}

// Equation 3: (Mat\_2 \* 5) - (Mat\_1 - 2)

System.out.println("\nResult of (Mat\_2 \* 5) - (Mat\_1 - 2):");

int[][] result3 = new int[3][3];

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

result3[i][j] = (Mat\_2[i][j] \* 5) - (Mat\_1[i][j] - 2);

System.out.print(result3[i][j] + "\t");

}

System.out.println();

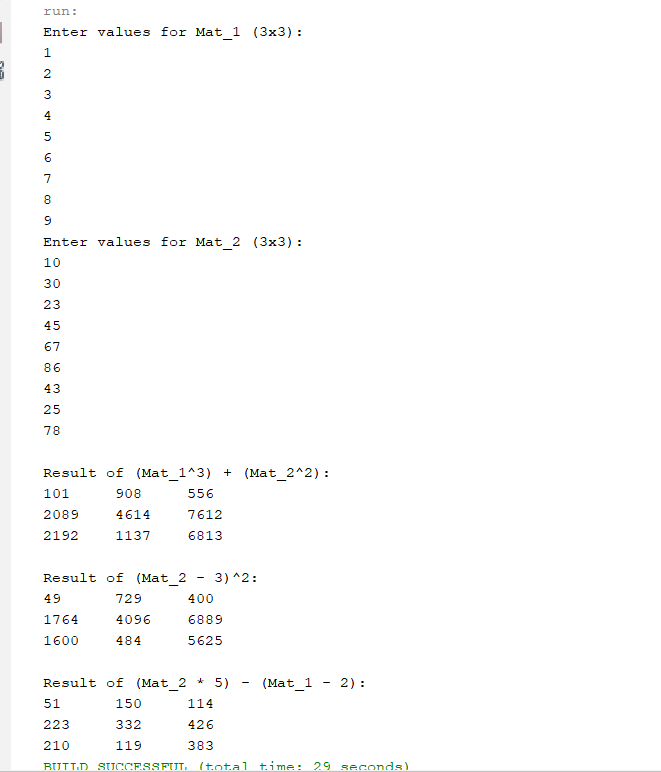
}

}

}

}

Output:



Exercise 4 (Recursion) (prodcut.java)

Write a recursive method to get multiply of all number from 1 up to given number. E.g. no. = 5 Result must be sum (1\*2\*3\*4\*5)

Source Code:

package product;

import java.util.Scanner;

public class Prodcut {

public static int multiplyUpTo(int num) {

if (num == 1) {

return 1;

}

return num \* multiplyUpTo(num - 1);

}

public static void main(String[] args) {

try (Scanner scanner = new Scanner(System.in)) {

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

int number = scanner.nextInt();

int result = multiplyUpTo(number);

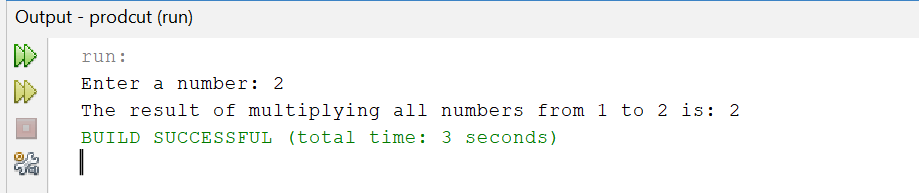
System.out.println("The result of multiplying all numbers from 1 to " + number + " is: " + result);

}

}

}

Output:



Exercise 5 (Recursion) (NumberSum.java)

Write a recursive function that takes two int as arguments and compute the sum of all number between provided two positive integers for example

If 1 and 20 are passed to the function answer should be 210.

Source Code:

package numbersum;

import java.util.Scanner;

public class NumberSum {

// Recursive function to compute the sum of numbers between start and end

public static int sumBetween(int start, int end) {

if (start > end) {

return 0;

}

return start + sumBetween(start + 1, end);

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input two positive integers from the user

System.out.print("Enter the starting number: ");

int start = scanner.nextInt();

System.out.print("Enter the ending number: ");

int end = scanner.nextInt();

// Call the recursive function and get the result

int result = sumBetween(start, end);

// Output the result

System.out.println("The sum of numbers between " + start + " and " + end + " is: " + result);

scanner.close();

}

}

Output:

