1. **Write a program to design a simple calculator using DatagramSocket and DatagramPacket classes demonstrating the basic mathematical arithmetic calculations.(Make necessary assumptions)**

**Code:**

**CalculatorServer.java:**

**package** edu.siescoms.dg2;

**import** java.net.\*;

**public** **class** CalculatorServer {

**public** **static** **void** main(String[] args) {

DatagramSocket socket = **null**;

**try** {

socket = **new** DatagramSocket(12345); // Port for communication

**while** (**true**) {

**byte**[] receiveData = **new** **byte**[1024];

DatagramPacket receivePacket = **new** DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String expression = **new** String(receivePacket.getData(), 0, receivePacket.getLength());

String[] parts = expression.split(" "); // Assuming the client sends "operand operator operand"

**if** (parts.length != 3) {

// Handle invalid requests

String errorMessage = "Invalid request";

DatagramPacket responsePacket = **new** DatagramPacket(errorMessage.getBytes(), errorMessage.length(), receivePacket.getAddress(), receivePacket.getPort());

socket.send(responsePacket);

**continue**;

}

**double** operand1 = Double.*parseDouble*(parts[0]);

**double** operand2 = Double.*parseDouble*(parts[2]);

**double** result = 0.0;

**switch** (parts[1]) {

**case** "+":

result = operand1 + operand2;

**break**;

**case** "-":

result = operand1 - operand2;

**break**;

**case** "\*":

result = operand1 \* operand2;

**break**;

**case** "/":

**if** (operand2 != 0) {

result = operand1 / operand2;

} **else** {

// Handle division by zero

String errorMessage = "Division by zero is not allowed";

DatagramPacket responsePacket = **new** DatagramPacket(errorMessage.getBytes(), errorMessage.length(), receivePacket.getAddress(), receivePacket.getPort());

socket.send(responsePacket);

**continue**;

}

**break**;

**default**:

// Handle invalid operator

String errorMessage = "Invalid operator";

DatagramPacket responsePacket = **new** DatagramPacket(errorMessage.getBytes(), errorMessage.length(), receivePacket.getAddress(), receivePacket.getPort());

socket.send(responsePacket);

**continue**;

}

String response = Double.*toString*(result);

DatagramPacket responsePacket = **new** DatagramPacket(response.getBytes(), response.length(), receivePacket.getAddress(), receivePacket.getPort());

socket.send(responsePacket);

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**if** (socket != **null** && !socket.isClosed()) {

socket.close();

}

}

}

}

**CalculatorClient.java:**

package edu.siescoms.dg2;

import java.io.IOException;

import java.net.\*;

import java.util.Scanner;

public class CalculatorClient {

public static void main(String[] args) {

DatagramSocket socket = null;

Scanner scanner = new Scanner(System.in);

try {

socket = new DatagramSocket();

InetAddress serverAddress = InetAddress.getByName("localhost"); // Change to the server's address if needed

int serverPort = 12345; // Match the server's port

System.out.println("Simple Calculator Client");

System.out.println("Available operations: +, -, \*, /");

System.out.println("Type 'exit' to quit.");

while (true) {

System.out.print("Enter an arithmetic expression: ");

String input = scanner.nextLine();

if (input.equalsIgnoreCase("exit")) {

break;

}

byte[] requestBytes = input.getBytes();

DatagramPacket requestPacket = new DatagramPacket(requestBytes, requestBytes.length, serverAddress, serverPort);

socket.send(requestPacket);

byte[] receiveData = new byte[1024];

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String response = new String(receivePacket.getData(), 0, receivePacket.getLength());

System.out.println("Result: " + response);

}

} catch (SocketException se) {

se.printStackTrace();

} catch (UnknownHostException uhe) {

uhe.printStackTrace();

} catch (IOException ioe) {

ioe.printStackTrace();

} finally {

if (socket != null && !socket.isClosed()) {

socket.close();

}

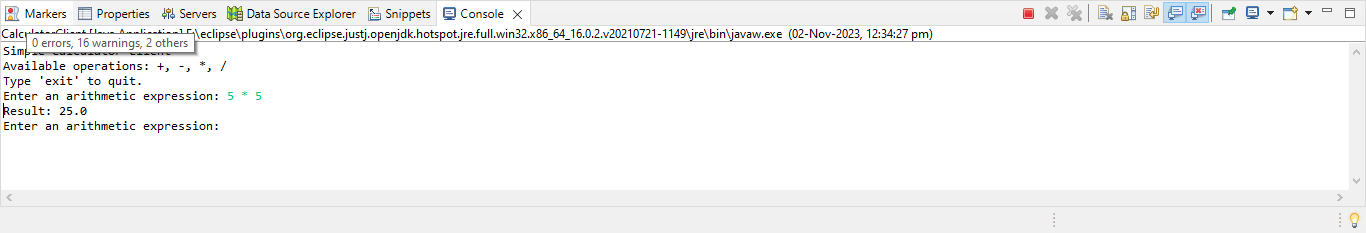
scanner.close();

}

}

}

**Output:**



1. **Write a program to design a Date Server using DatagramSocket and DatagramPacket classes which receives a message from client to get today’s date and time. (Make necessary assumptions).**

**Code:**

**DateServer.java:**

package edu.siescoms.dg2;

import java.net.\*;

import java.util.Date;

public class DateServer {

public static void main(String[] args) {

DatagramSocket socket = null;

try {

socket = new DatagramSocket(12345); // Port for communication

while (true) {

byte[] receiveData = new byte[1024];

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String request = new String(receivePacket.getData(), 0, receivePacket.getLength());

if (request.equals("get\_date\_time")) {

// Get the current date and time

String currentDateTime = new Date().toString();

byte[] responseBytes = currentDateTime.getBytes();

// Create a response packet and send it back to the client

DatagramPacket responsePacket = new DatagramPacket(responseBytes, responseBytes.length, receivePacket.getAddress(), receivePacket.getPort());

socket.send(responsePacket);

} else {

// Handle invalid requests

String errorMessage = "Invalid request";

byte[] errorBytes = errorMessage.getBytes();

DatagramPacket responsePacket = new DatagramPacket(errorBytes, errorBytes.length, receivePacket.getAddress(), receivePacket.getPort());

socket.send(responsePacket);

}

}

} catch (Exception e) {

e.printStackTrace();

} finally {

if (socket != null && !socket.isClosed()) {

socket.close();

}

}

}

}

**DateClient.java:**

**package** edu.siescoms.dg2;

**import** java.net.\*;

**public** **class** DateClient {

**public** **static** **void** main(String[] args) {

DatagramSocket socket = **null**;

**try** {

socket = **new** DatagramSocket();

InetAddress serverAddress = InetAddress.*getByName*("localhost"); // Change to the server's address if needed

**int** serverPort = 12345; // Match the server's port

String request = "get\_date\_time";

**byte**[] requestBytes = request.getBytes();

DatagramPacket requestPacket = **new** DatagramPacket(requestBytes, requestBytes.length, serverAddress, serverPort);

socket.send(requestPacket);

**byte**[] receiveData = **new** **byte**[1024];

DatagramPacket receivePacket = **new** DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String response = **new** String(receivePacket.getData(), 0, receivePacket.getLength());

System.***out***.println("Received date and time: " + response);

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**if** (socket != **null** && !socket.isClosed()) {

socket.close();

}

}

}

}

**Output:**

