**Guru Ghasidas Vishwavidyalaya, Bilaspur (C.G.)**

**A picture containing text, gear

Description automatically generated**

**Department of Computer Science & Information Technology**

**Course name - MCA**

**Advance Java Lab File**

|  |  |
| --- | --- |
| **Guided By :-** | **Presented By :-** |
| **Mr. Suman Laha** | **Adarsh Chakraborty** |
|  | **GGV/21/05004** |
|  | **Roll no. – 21072104** |

**INDEX PAGE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No. | Name of Program | Page No. | Date of Project | Remark |
| 1. | Write a Java program to demonstrate & understand the differences of String, StringBuffer & StringBuilder. |  |  |  |
| 2. | Write a Java program to demonstrate & understand at least 15 different methods from String, StringBuffer & StringBuilder. |  |  |  |
| 3. | Write a Java program to design a class “Complex” which represents complex numbers and perform operations on complex numbers. Test the Complex class on addition, subtraction, multiplication & division of complex numbers. |  |  |  |
| 4. | Write a Java program to Design a class “FileAccess” which manages different operations on text files having UTF-8/ UTF-16 formats:  a) read: reads a portion of the file as per the parameters.  b) write: writes text into the file at specific position with optional overwrite instruction.  c) copy: copies the file |  |  |  |
| 5. | Write a Java program to copy multiple “csv” files into single one. |  |  |  |
| 6. | Utilizing nio, write a Java program to read all lines of a file into a collection and delete those lines from the collection which are containing a given string as a substring and write back the remaining lines from collection into the file to update the file. |  |  |  |
| 7. | Utilizing nio, write a Java program to read all lines of a file into a collection and delete those lines from the collection which are containing a given string as a substring and write back the remaining lines from collection into the file to update the file. |  |  |  |
| 8. | Utilizing nio, write a Java program to write all non redundant lines to a file continuously taken from user at prompt until user writes “exit”. |  |  |  |
| 9. | Write a Java program to perform queries from the sales record stored in a data file as follows: salesman<>product<>sale amount<>sale date Where <> is field separator and new line is record separator. Find,  a) highest selling product  b) highest selling salesman  c) total sale in the month of November-2021 |  |  |  |
| 10. | Write a Java program to receive n number of names from user and keep them into a file and perform the operations below.  i. Search name by key.  ii. Find and replace name without disturbing the position of the name in the file.  iii. Remove the duplicate name from the file. |  |  |  |
| 11. | Write a Java program to print a sequence of numbers by three threads. |  |  |  |
| 12. | Write a Java program to create three threads such that  (A) ThreadA prints 1 to 50. (B)ThreadB starts only after when ThreadA prints 10 and prints 10.1 to 10.50. (C) ThreadC starts only after when ThreadB prints 10.20.1 to 10.20.50. |  |  |  |
| 13. | There are 50 strings in a shared collection. Write a Java program to create two threads which are deleting a line from the collection at a time. All threads should remain alive until the collection is not empty. |  |  |  |
| 14. | Write a Java program to create two threads. First thread appends a line of text into a file at a time and is alive till 5 lines are appended. Second thread displays the lines from the file at a time and goes off if all 5 lines are displayed at a time. |  |  |  |
| 15. | Write a JDBC program to connect to a database and demonstrate the Statement. |  |  |  |
| 16. | Write a JDBC program to connect to a database and demonstrate the PreparedStatement. |  |  |  |
| 17. | Write a JDBC program to connect to a database and demonstrate the CallableStatement. |  |  |  |
| 18. | Write a Servlet program to connect to a database and generate the page contains Tables combo, Fields combo and data area:  Table combo contains all tables of the database.On selection of table combo Field combo contains all fields of selected table. On selection of field combo all data of selected field are to be displayed on the data area. If we select all fields from Field combo then respective table data are to be displayed. |  |  |  |
| 19. | Implement Session tracking in servlet using:  a. Hidden from field  b. Url re-write  c. Cookies  d. Session |  |  |  |
| 20. | Write a Servlet program to demonstrate initial parameter access from a servlet page. |  |  |  |
| 21. | Write a Servlet program to demonstrate parameter access from a servlet page. |  |  |  |
| 22. | Write a JSP program to demonstrate access bean object in different scopes. How to access bean object which was opened in a different page? |  |  |  |
| 23. | Write a JSP program to demonstrate at least five action tags. |  |  |  |
| 24. | Write a JSP program to demonstrate directives. |  |  |  |
| 25. | Write a JSP program to demonstrate initial parameter and request parameters access from a JSP page. |  |  |  |

**Problem 1: Write a Java program to demonstrate & understand the differences of String, StringBuffer & StringBuilder.**

// Java program to demonstrate difference between

// String, StringBuilder and StringBuffer

// Main class

class Main {

// Concatenates to String

public static void concat1(String s1)

{

s1 = s1 + " Chakraborty";

}

// Method 2

// Concatenates to StringBuilder

public static void concat2(StringBuilder s2)

{

s2.append(" Chakraborty");

}

// Method 3

// Concatenates to StringBuffer

public static void concat3(StringBuffer s3)

{

s3.append(" Chakraborty");

}

// Method 4

// Main driver method

public static void main(String[] args)

{

// Custom input string

// String 1

String s1 = "Adarsh";

// Calling above defined method

concat1(s1);

// s1 is not changed

System.out.println("String: " + s1 );

// String 2

StringBuilder s2 = new StringBuilder("Adarsh");

// Calling above defined method

concat2(s2);

// s2 is changed

System.out.println("StringBuilder: " + s2 );

// String 3

StringBuffer s3 = new StringBuffer("Adarsh");

concat3(s3);

// s3 is changed

System.out.println("StringBuffer: " + s3 );

System.out.println("\n Now, Checking time comparision, appending x1000 times.\n");

long startTime1 = System.currentTimeMillis();

for (int i=0; i<10000; i++){

concat1(s1);

}

System.out.println("Time taken by String Class: " + (System.currentTimeMillis() - startTime1) + "ms");

long startTime2 = System.currentTimeMillis();

for (int i=0; i<10000; i++){

concat2(s2);

}

System.out.println("Time taken by StringBuilder: " + (System.currentTimeMillis() - startTime2) + "ms");

long startTime3 = System.currentTimeMillis();

for (int i=0; i<10000; i++){

concat3(s3);

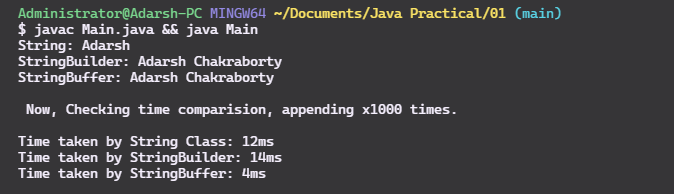
}

System.out.println("Time taken by StringBuffer: " + (System.currentTimeMillis() - startTime3) + "ms");

}

}

**Output – 1**



**Problem 2: Write a Java program to demonstrate & understand at least 15 different methods from String, StringBuffer & StringBuilder.**

class Main {

public static void main(String[] args) {

String s1 = " Adarsh Chakraborty ";

// 1. trim - Removes whitespaces

System.out.println("1. trim() Method:" +s1.trim());

// 2. toUpperCase

System.out.println("2. Uppercased: " + s1.toUpperCase());

// 3. toLowerCase

System.out.println("3. Lowercased: "+ s1.toLowerCase());

// 4. isEmpty

System.out.println("4. Is String Empty?: "+ s1.isEmpty());

// 5. length

System.out.println("5. Length of s1 string: "+ s1.length());

StringBuilder s2 = new StringBuilder("Adarsh Chakraborty");

// 6. append()

s2.append(" MCA-1");

System.out.println("6. after append: "+ s2);

// 7. insert

s2.insert(s2.length(), " 2021-22");

System.out.println("7. after insert: "+ s2);

// 8. replace

s2.replace(0, s2.length(), "Adarsh Chakraborty");

System.out.println("8. Replaced whole string: " + s2);

// 9. delete

s2.delete(0, 6);

System.out.println("9. Deleted from [0 - 6] index: " + s2);

// 10. reverse

s2.reverse();

System.out.println("10. Reversed String: " + s2);

StringBuffer s3 = new StringBuffer("Adarsh Chakraborty");

// 11. capacity()

System.out.println("11. Capacity of s3: " + s3.capacity());

// 12. charAt()

System.out.println("12. Char at Index 0: " + s3.charAt(0));

// 13. indexOf()

System.out.println("13. Index of 'Chak' in given string: " +s3.indexOf("Chak"));

// 14. SubString

String substr = s3.substring(0,6);

System.out.println("14. Substring from index [0-6]: " + substr);

// 15. deleteCharAt

System.out.println("\* String s3: " + s3);

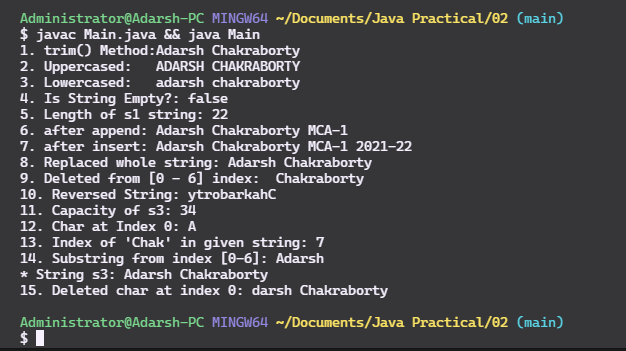
s3.deleteCharAt(0);

System.out.println("15. Deleted char at index 0: " + s3);

}

}

**Output- 2**



**Problem 3: Write a Java program to design a class “Complex” which represents complex numbers and perform operations on complex numbers. Test the Complex class on addition, subtraction, multiplication & division of complex numbers.**

public class Complex {

private double real;

private double imaginary;

public double getReal() {

return this.real;

}

public void setReal(double real) {

this.real = real;

}

public double getImaginary() {

return this.imaginary;

}

public void setImaginary(double imaginary) {

this.imaginary = imaginary;

}

public Complex() {

this.setReal(0);

this.setImaginary(0);

}

public Complex(double real, double imaginary){

this.setReal(real);

this.setImaginary(imaginary);

}

public void setRealAndImaginary(double real, double imaginary){

this.setReal(real);

this.setImaginary(imaginary);

}

@Override

public String toString() {

return "Values : [ (" + real + " ) + (" + imaginary + " i ) ]";

}

public Complex addition(Complex temp){

Complex result = new Complex();

result.real = this.getReal() + temp.getReal();

result.imaginary = this.getImaginary() + temp.getImaginary();

return result;

}

public Complex subtraction(Complex temp){

Complex result = new Complex();

result.real = this.getReal() - temp.getReal();

result.imaginary = this.getImaginary() - temp.getImaginary();

return result;

}

public Complex multiplication(Complex temp){

Complex result = new Complex();

result.real = ( this.getReal() \* temp.getReal() ) - ( this.getImaginary() \* temp.getImaginary());

result.imaginary = ( this.getReal() \* temp.getImaginary() ) + ( this.getImaginary() \* temp.getReal());

return result;

}

public Complex division(Complex temp){

Complex result = new Complex();

result.real = ((( this.getReal() \* temp.getReal() ) + ( this.getImaginary() \* temp.getImaginary())) / (Math.pow(temp.getReal(), 2) + Math.pow(temp.getImaginary(), 2)));

result.imaginary =((( this.getReal() \* temp.getImaginary() ) - ( this.getImaginary() \* temp.getReal() )) / (Math.pow(temp.getReal(), 2) + Math.pow(temp.getImaginary(), 2)));

return result;

}

public static void main(String[] args) {

Complex complex = new Complex();

Complex complex1 = new Complex(2,-1);

Complex complex2 = new Complex(3,4);

System.out.println("complex " + complex);

System.out.println("complex1 " + complex1);

System.out.println("complex2 " + complex2);

System.out.println("\n\n\nAddition " + complex1.addition(complex2));

System.out.println("Subtraction " + complex1.subtraction(complex2));

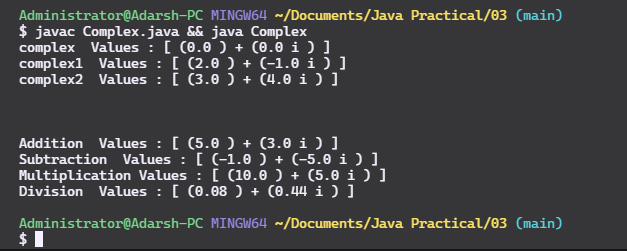
System.out.println("Multiplication " + complex1.multiplication(complex2));

System.out.println("Division " + complex1.division(complex2));

}

}

**Output - 3**



**Problem 4: Write a Java program to Design a class “FileAccess” which manages different operations on text files having UTF-8/ UTF-16 formats:**

**a) read: reads a portion of the file as per the parameters.**

**b) write: writes text into the file at specific position with optional overwrite instruction.**

**c) copy: copies the file**

import java.nio.file.Files;

import java.nio.file.Paths;

import java.io.\*;

import java.util.Scanner;

class FileRead {

String fileName;

FileRead(String f){

this.fileName = f;

}

void readLine(int n) {

String line;

try {

BufferedReader br = new BufferedReader(new FileReader(fileName));

for (int i = 0; i < n; i++)

br.readLine();

line = br.readLine();

System.out.println(line);

}

catch(IOException e){

System.out.println(e);

}

}

void readUntill(int n) {

String line;

try {

BufferedReader br = new BufferedReader(new FileReader(fileName));

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

line = br.readLine();

System.out.println(line);

}

}

catch(IOException e){

System.out.println(e);

}

}

}

class FileWrite {

File file;

FileWrite(String fname){

file = new File(fname);

}

void writetoFile(String content, int position){

// No overwrite

// Read entire file

// Put somethings at a specific position

// Put the data back in the file

try{

String fName = file.getName();

StringBuilder fileContent = new StringBuilder(Files.readString(Paths.get(fName)));

fileContent.insert(position, content);

Files.write( Paths.get(fName), fileContent.toString().getBytes());

}

catch(IOException e){

System.out.println(e);

}

}

void writetoFile(String content, int position, Boolean overwrite){

String fName = file.getName();

try{

if(overwrite == true){

// file, appendMode?

Files.write( Paths.get(fName), content.getBytes());

}else{

StringBuilder fileContent = new StringBuilder(Files.readString(Paths.get(fName)));

fileContent.insert(position, content);

Files.write( Paths.get(fName), fileContent.toString().getBytes());

}

}

catch(Exception e){

System.out.println(e);

}

}

}

class FileCopy{

File src;

FileCopy(String f){

this.src = new File(f);

}

void copyTo(String fileName){

try{

File dest = new File(fileName);

// using copy(InputStream,Path Target); method

Files.deleteIfExists(dest.toPath());

Files.copy(src.toPath(), dest.toPath());

}catch(Exception e){

System.out.println(e);

}

}

}

class FileAccess {

public static void main(String args[]){

// Read a portion of file.

FileRead file = new FileRead("input.txt");

// Pass line number to read.

file.readUntill(3);

file.readLine(4);

// Write to a file at specific position.

FileWrite file1 = new FileWrite("File1.txt");

FileWrite file2 = new FileWrite("File2.txt");

// Content, LineNumber (Position), Overwrite? (optional)

file1.writetoFile("Adarsh Chakraborty", 4);

file2.writetoFile("Hello, This file was overwritten.", 0, true );

// Copies input.txt to output.txt

FileCopy file3 = new FileCopy("input.txt");

file3.copyTo("output.txt");

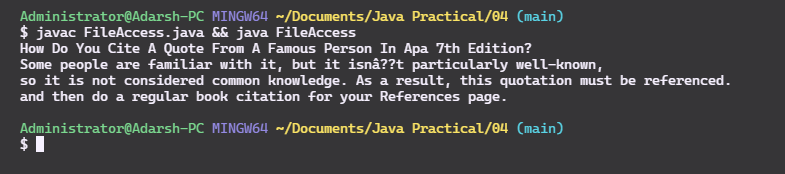
}

}

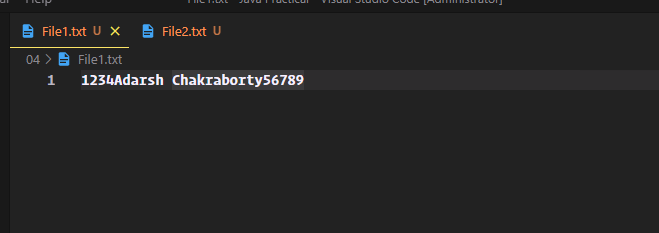
**Output – 4**

**// Reading a specific portion of file**

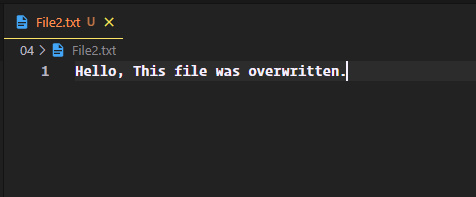
**// Reading untill line 3 and line 4**



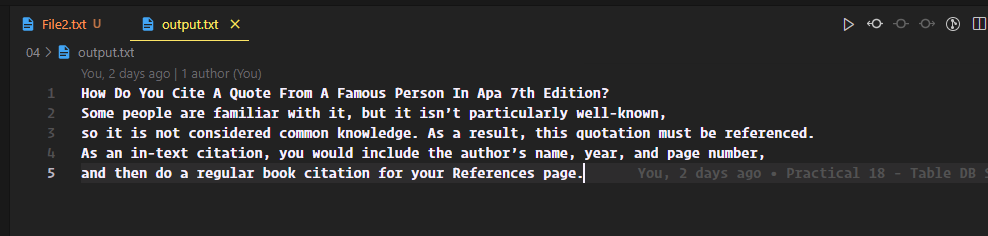
**// Inserting text at specific position**



**// Overwrite**



**// input.txt file copied to output.txt**



**Problem 5: Write a Java program to copy multiple “csv” files into single one.**

//MergeDemo.java

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

//import java.util.Arrays;

import java.util.Iterator;

import java.util.Scanner;

public class MergeDemo {

public static void main(String[] args) {

String idFile = "100";

int numFiles = 3;

try {

mergeCsvFiles(idFile, numFiles);

} catch (IOException e) {

e.printStackTrace();

}

}

private static void mergeCsvFiles(String idFile, int numFiles) throws IOException {

ArrayList<File> files = new ArrayList<File>();

Iterator<File> iterFiles;

File fileOutput;

BufferedWriter fileWriter;

BufferedReader fileReader;

String csvFile;

String csvFinal = "./output/merged.csv";

String[] headers = null;

String header = null;

// Files: Input

for (int i = 1; i <= numFiles; i++) {

csvFile = "./csvfiles/cities" + "\_" + i + ".csv";

files.add(new File(csvFile));

}

// Files: Output

fileOutput = new File(csvFinal);

if (fileOutput.exists()) {

fileOutput.delete();

}

try {

fileOutput.createNewFile();

} catch (IOException e) {

}

iterFiles = files.iterator();

fileWriter = new BufferedWriter(new FileWriter(csvFinal, true));

// Headers

Scanner scanner = new Scanner(files.get(0));

if (scanner.hasNextLine())

header = scanner.nextLine();

scanner.close();

fileWriter.write(header);

fileWriter.newLine();

while (iterFiles.hasNext()) {

String line;// = null;

String[] firstLine;// = null;

File nextFile = iterFiles.next();

fileReader = new BufferedReader(new FileReader(nextFile));

if ((line = fileReader.readLine()) != null)

firstLine = line.split(";");

while ((line = fileReader.readLine()) != null) {

fileWriter.write(line);

fileWriter.newLine();

}

fileReader.close();

}

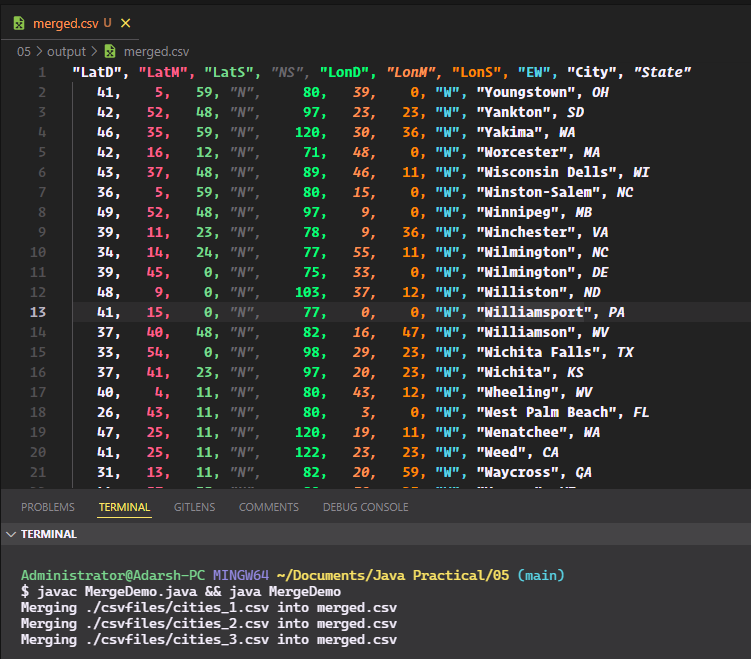
fileWriter.close();

}

}

Output – 5

// Merged Csv file



**Problem 6: Utilizing nio, write a Java program to read all lines of a file into a collection and delete those lines from the collection which are containing a given string as a substring and write back the remaining lines from collection into the file to update the file.**

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.List;

import java.nio.charset.Charset;

class Main {

public static void main (String args []) throws IOException{

String[] findAndRemoveCity = {"Chennai","Kolkata","Surat ","Pune","Jaipur","Kanpur","Nagpur","Indore","Thane","Bhopal","Patna","Vadodara","Ghaziabad","Ludhiana"};

Path filePath = Paths.get("data.txt");

List<String> lines = Files.readAllLines(filePath);

for(String city : findAndRemoveCity){

// Check for substring here

lines.remove(city);

}

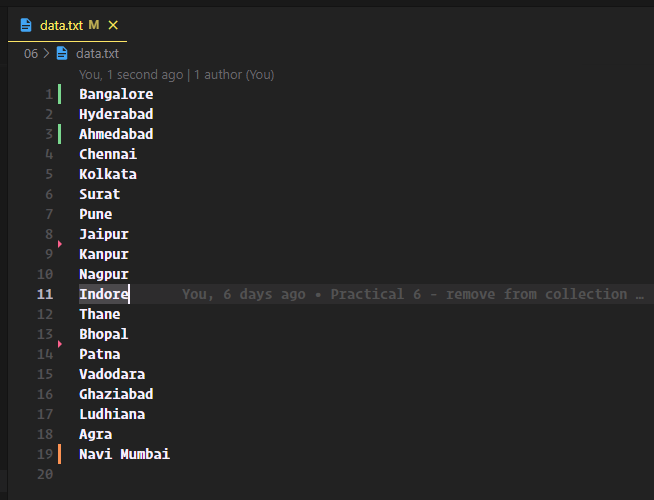
Files.write(filePath, lines, Charset.forName("UTF-8"));

}

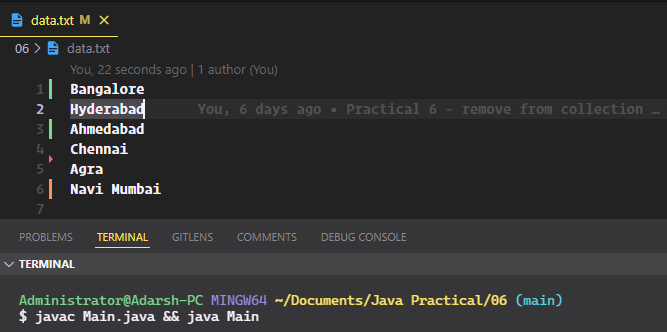
}

**Output – 6**

**// Original data.txt file**



**// Given cities removed from collection and file.**



**Problem 7: Utilizing nio, write a Java program to read all lines of a file into a collection and delete those lines from the collection which are containing a given string as a substring and write back the remaining lines from collection into the file to update the file.**

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.List;

import java.nio.charset.Charset;

class Main {

public static void main (String args []) throws IOException{

String[] findAndRemoveCity = {"Chennai","Kolkata","Surat ","Pune","Jaipur","Kanpur","Nagpur","Indore","Thane","Bhopal","Patna","Vadodara","Ghaziabad","Ludhiana"};

Path filePath = Paths.get("data.txt");

List<String> lines = Files.readAllLines(filePath);

for(String city : findAndRemoveCity){

// Check for substring here

lines.remove(city);

}

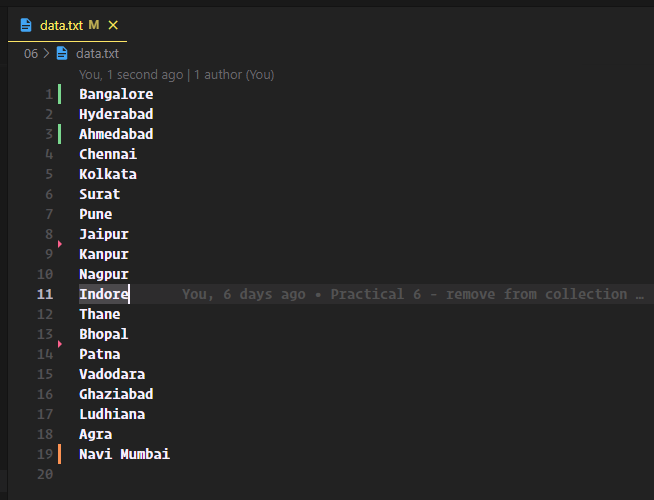
Files.write(filePath, lines, Charset.forName("UTF-8"));

}

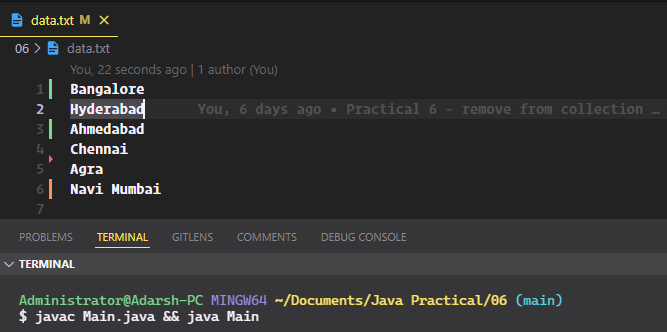
}

**Output – 7**

**// Original data.txt file**



**// Given cities removed from collection and file.**



**Problem 8: Utilizing nio, write a Java program to write all non redundant lines to a file continuously taken from user at prompt until user writes “exit”.**

import java.util.Scanner;

import java.nio.\*;

import java.nio.file.Paths;

import java.nio.file.Path;

import java.nio.file.Files;

import java.io.IOException;

import java.nio.file.StandardOpenOption;

import java.util.HashMap;

class Main {

public static void main(String args[]) throws IOException{

Scanner sc = new Scanner(System.in);

HashMap<String,String> inputHistroy = new HashMap<String,String>();

System.out.println("Hint: type exit to exit anytime.");

System.out.println("Enter file name:");

String fileName = sc.nextLine();

Path newFilePath = Paths.get(fileName + ".txt");

Files.deleteIfExists(newFilePath);

Files.createFile(newFilePath);

System.out.println("Enter Data for the file: (Type exit to exit anytime).");

while(true){

String input = sc.nextLine();

if(input.equals("exit")) break;

if(inputHistroy.containsKey(input.toLowerCase())){

System.out.println("Only Non-redundant lines allowed.");

continue;

}

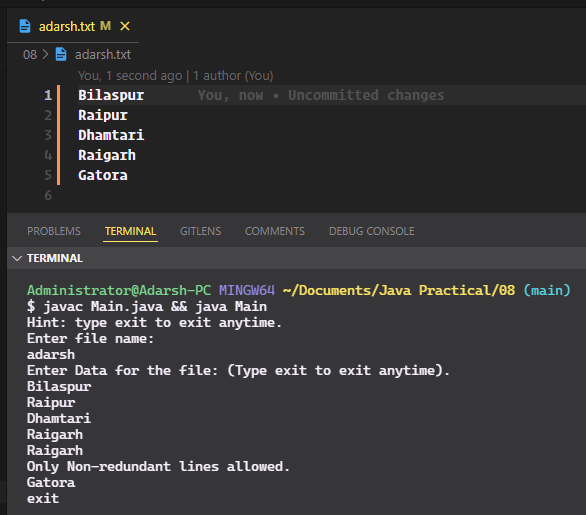
inputHistroy.put(input.toLowerCase(),input);

Files.writeString(newFilePath, input+"\n", StandardOpenOption.APPEND);

}

}

}



**Problem 9: Write a Java program to perform queries from the sales record stored in a data file as follows: salesman<>product<>sale amount<>sale date Where <> is field separator and new line is record separator. Find,**

**a) highest selling product**

**b) highest selling salesman**

**c) total sale in the month of November-2021.**

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.List;

import java.util.ArrayList;

import java.time.LocalDate;

import java.time.YearMonth;

import java.time.format.DateTimeFormatter;

import java.util.HashMap;

import java.util.Map;

import java.util.Collections;;

class Main{

public static void main(String args[]) throws IOException{

Path filePath = Paths.get("records.txt");

List<String> salesRecords = Files.readAllLines(filePath);

List<Sale> Sales = new ArrayList<Sale>();

for(String saleRecord : salesRecords){

Sales.add(new Sale(saleRecord));

}

// Now we have all the data in Sales List.

// Product Name, Count;

HashMap<String, Integer> hsProducts = new HashMap<String, Integer>();

HashMap<String, Integer> hsSalesman = new HashMap<String, Integer>();

HashMap<YearMonth, Integer> totalSale = new HashMap<YearMonth, Integer>();

// Find Records of this Year and month

YearMonth yearMonth = YearMonth.of(2021, 11);

int salesWorthInTheMonth = 0;

for(Sale sale : Sales){

Sale highestSellingProduct,highestSellingSalesman;

if(hsProducts.containsKey(sale.product)){

int currentCount = hsProducts.get(sale.product);

currentCount++;

hsProducts.put(sale.product,currentCount);

}else{

hsProducts.put(sale.product,1);

}

if(hsSalesman.containsKey(sale.salesman)){

int currentCount = hsSalesman.get(sale.salesman);

currentCount++;

hsSalesman.put(sale.salesman,currentCount);

}else{

hsSalesman.put(sale.salesman,1);

}

if(YearMonth.from(sale.date).equals(yearMonth)){

if(totalSale.containsKey(yearMonth)){

int currentCount = totalSale.get(yearMonth);

currentCount++;

totalSale.put(yearMonth,currentCount);

salesWorthInTheMonth += Integer.parseInt(sale.amount);

}else{

totalSale.put(yearMonth,1);

salesWorthInTheMonth += Integer.parseInt(sale.amount);

}

}

}

String highestSellingProduct = Collections.max(hsProducts.entrySet(), Map.Entry.comparingByValue()).getKey();

String highestSellingSalesman = Collections.max(hsSalesman.entrySet(), Map.Entry.comparingByValue()).getKey();

YearMonth totalSaleKey = Collections.max(totalSale.entrySet(), Map.Entry.comparingByValue()).getKey();

System.out.println("Total Sales:\n" + hsProducts.toString());

System.out.println("Highest Selling Product: "+ highestSellingProduct + " with total " + hsProducts.get(highestSellingProduct) + " sales.");

System.out.println("Highest Selling Salesman: "+ highestSellingSalesman + " with total " + hsSalesman.get(highestSellingSalesman) + " sales.");

System.out.println("Total Sales in "+ yearMonth.getMonth() + "-" + yearMonth.getYear() + ": "+ totalSale.get(totalSaleKey) + " (Amount: " + salesWorthInTheMonth+")");

}

}

class Sale {

String salesman, product, amount;

LocalDate date;

Sale(String record) {

String[] records = record.split("<>");

this.salesman = records[0];

this.product = records[1];

this.amount = records[2];

this.date = LocalDate.parse(records[3]);

//Parsing the given String to Date object

}

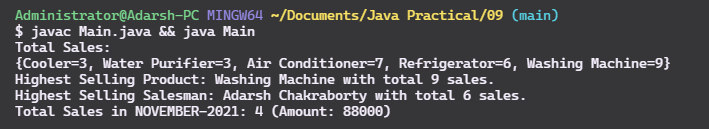
public String toString() {

return "Salesman: '" + this.salesman + "', Product: '" + this.product + "', Amount: '" + this.amount + "'" + "', Date: '" + this.date + "'";

}

}

**Output – 9**



**Problem 10: Write a Java program to receive n number of names from user and keep them into a file and perform the operations below.**

**i. Search name by key.**

**ii. Find and replace name without disturbing the position of the name in file.**

**iii. Remove the duplicate name from the file.**

import java.io.IOException;

import java.util.Scanner;

import java.nio.\*;

import java.nio.file.Paths;

import java.nio.file.Path;

import java.nio.file.Files;

import java.nio.charset.Charset;

import java.nio.file.StandardOpenOption;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Set;

import java.util.List;

class Main {

public static HashMap<Integer,String> inputHistroy = new HashMap<Integer,String>();

public static void main(String args[]) throws IOException{

Path newFilePath = Paths.get("names.txt");

Scanner sc = new Scanner(System.in);

int N = 5;

System.out.println("How many names you want to Enter?: [Integer]");

try {

N = sc.nextInt();

}catch(Exception e){

System.out.println("Error: Integer was expected: [1-3]");

}

Files.deleteIfExists(newFilePath);

Files.createFile(newFilePath);

System.out.println("Enter "+ N + " names:");

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

String input = sc.nextLine();

inputHistroy.put(i,input);

Files.writeString(newFilePath, input+"\n", StandardOpenOption.APPEND);

}

showSelection();

}

static void showSelection(){

Scanner sc = new Scanner(System.in);

int selection = 0;

System.out.println("\n\nEnter selection:\n1.Search Name by Key.\n2.Find and Replace name.\n3.Remove dublicates names from file.");

try {

selection = sc.nextInt();

}catch(Exception e){

System.out.println("Error: Integer was expected: [1-3]");

}

switch (selection){

case 1:

searchName();

break;

case 2:

findAndReplace();

break;

case 3:

removeDublicateNames();

break;

default:

System.out.println("Invalid selection, Enter [1-3]:");

showSelection();

}

}

static void searchName() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter key to search:");

int selection = 0;

try {

selection = sc.nextInt();

}catch(Exception e){

System.out.println("Error: Integer was expected:");

}

String result = inputHistroy.get(selection);

if(result == null){

System.out.println("No name found with that Key.");

}else{

System.out.println("\nSearch result: " + result);

}

System.out.println("Press enter to go back to main menu...");

try{

System.in.read();

}catch(IOException e){

e.printStackTrace();

}

showSelection();

}

static void findAndReplace(){

try{

Scanner sc = new Scanner(System.in);

System.out.println("\n|| Find and Replace ||\nEnter the name to find:");

String findName = sc.nextLine();

System.out.println("Enter names to replace with:");

String replaceName = sc.nextLine();

Path filePath = Paths.get("names.txt");

List<String> lines = Files.readAllLines(filePath);

int index = 0;

int changes = 0;

for(String name : lines){

if(name.toLowerCase().equals(findName.toLowerCase())){

System.out.println(name);

lines.set(index, replaceName);

changes++;

}

index++;

}

Files.write(filePath, lines, Charset.forName("UTF-8"));

System.out.println("Task completed. "+ changes + " names were replaced.");

System.out.println("Press enter to go back to main menu...");

try{

System.in.read();

}catch(IOException e){

e.printStackTrace();

}

showSelection();

}catch(Exception e){

}

}

static void removeDublicateNames(){

try{

System.out.println("Removing all dublicate names from the records, Please wait...");

Path filePath = Paths.get("names.txt");

List<String> names = Files.readAllLines(filePath);

Set<String> uniqueNames = new HashSet<String>();

for (String name : names)

uniqueNames.add(name);

Files.write(filePath, uniqueNames, Charset.forName("UTF-8"));

System.out.println("Task completed. Press enter to go back to menu...");

System.in.read();

}catch(IOException e){

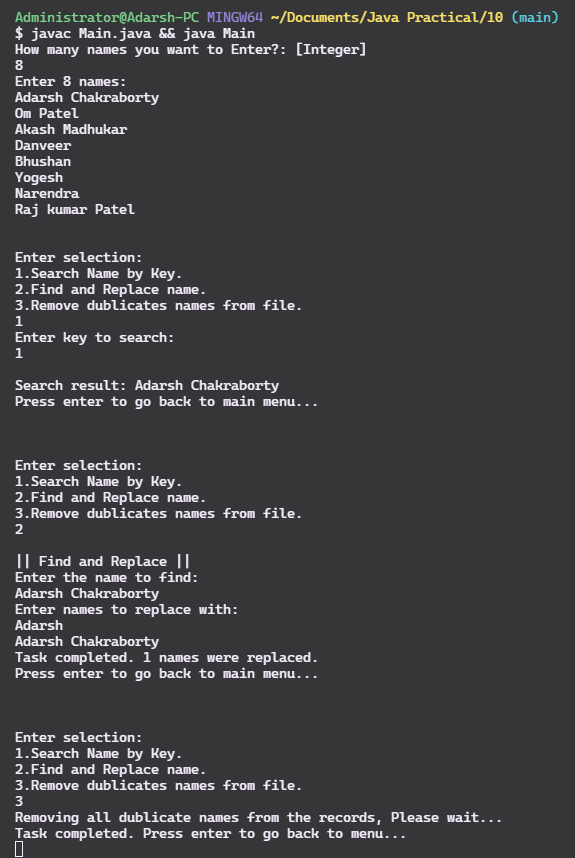
e.printStackTrace();

}

showSelection();

}

}

**Output - 10**

**Problem 11: Write a Java program to print a sequence of numbers by three threads.**

public class PrintNumbers {

final static int MAX\_NUMBERS = 25;

public static void main(String[] args) {

// shared object

PrintNumbers obj = new PrintNumbers();

// Creating 3 threads

Thread t1 = new Thread(new NumberRunnable(obj, 0), "T1");

Thread t2 = new Thread(new NumberRunnable(obj, 1), "T2");

Thread t3 = new Thread(new NumberRunnable(obj, 2), "T3");

t1.start();

t2.start();

t3.start();

}

}

class NumberRunnable implements Runnable{

PrintNumbers obj;

int threadNumber;

static int number = 0;

NumberRunnable(PrintNumbers obj, int result){

this.obj = obj;

this.threadNumber = result;

}

@Override

public void run() {

while (number < PrintNumbers.MAX\_NUMBERS) {

synchronized(obj) {

if(number % 3 == threadNumber && number < PrintNumbers.MAX\_NUMBERS){

System.out.println(Thread.currentThread().getName() + " - " + ++number);

}

}

}

}

}

**Output – 11**



**Problem 12: Write a Java program to create three threads such that**

**(A) ThreadA prints 1 to 50.**

**(B)ThreadB starts only after when ThreadA prints 10 and prints 10.1 to 10.50.**

**(C) ThreadC starts only after when ThreadB prints 10.20.1 to 10.20.50.**

import java.io.\*;

import java.util.\*;

// Class 1

// Thread1

// Helper class extending Thread class

public class PrintNumbers {

// Main driver method

public static void main(String[] args)

{

ResourceLock lock = new ResourceLock();

// Creating object(thread) of class 2

ThreadA obj1 = new ThreadA(lock);

ThreadB obj2 = new ThreadB(lock);

ThreadC obj3 = new ThreadC(lock);

// Starting the thread

obj1.start();

obj2.start();

obj3.start();

}

}

// Class 2

// Thread2

// Helper class extending Thread class

class ThreadA extends Thread{

ResourceLock lock;

ThreadA(ResourceLock lock){

this.lock = lock;

}

@Override

public void run() {

try{

synchronized (lock) {

for (int i = 1; i <= 50; i++) {

while(lock.flag !=1 ){

lock.wait();

}

System.out.println("ThreadA prints "+ i);

if(i == 10){

lock.flag = 2;

lock.notify();

}

}

}

}catch (Exception e) {

System.out.println("Exception 1 :"+e.getMessage());

}

}

}

class ThreadB extends Thread{

ResourceLock lock;

ThreadB(ResourceLock lock){

this.lock = lock;

}

@Override

public void run() {

try{

synchronized (lock) {

for (int i = 1; i <= 50; i++) {

while(lock.flag != 2){

lock.wait();

}

System.out.println("ThreadB prints 10."+ i);

if(i == 20){

lock.flag = 3;

lock.notifyAll();

}

}

lock.flag = 1;

lock.notify();

}

}catch (Exception e) {

System.out.println("Exception 2 :"+e.getMessage());

}

}

}

class ThreadC extends Thread{

ResourceLock lock;

ThreadC(ResourceLock lock){

this.lock = lock;

}

@Override

public void run() {

try{

synchronized (lock) {

for (int i = 1; i <= 50; i++) {

while(lock.flag != 3){

lock.wait();

}

System.out.println("ThreadC prints 10.20."+ i);

}

lock.flag = 2;

lock.notify();

}

}catch (Exception e) {

System.out.println("Exception 3 :"+e.getMessage());

}

}

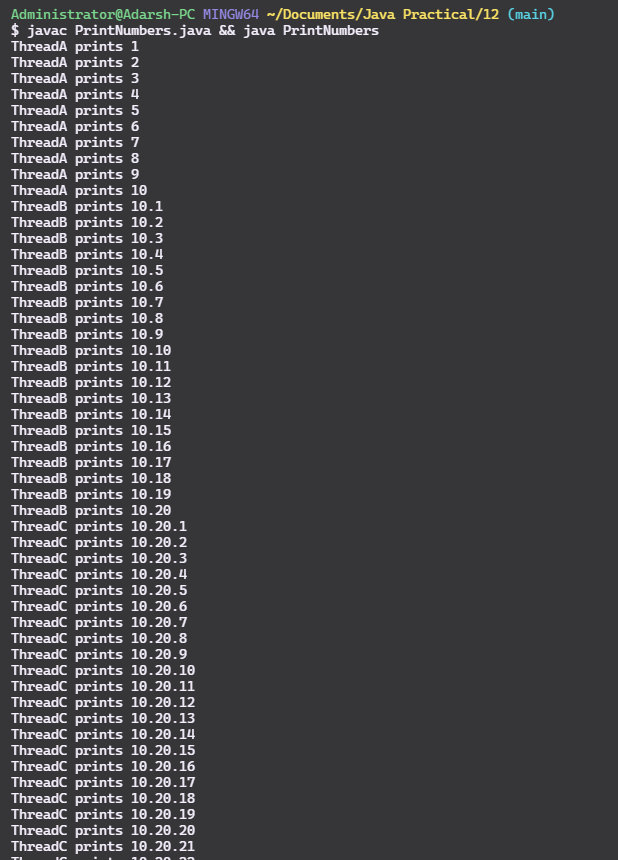
}

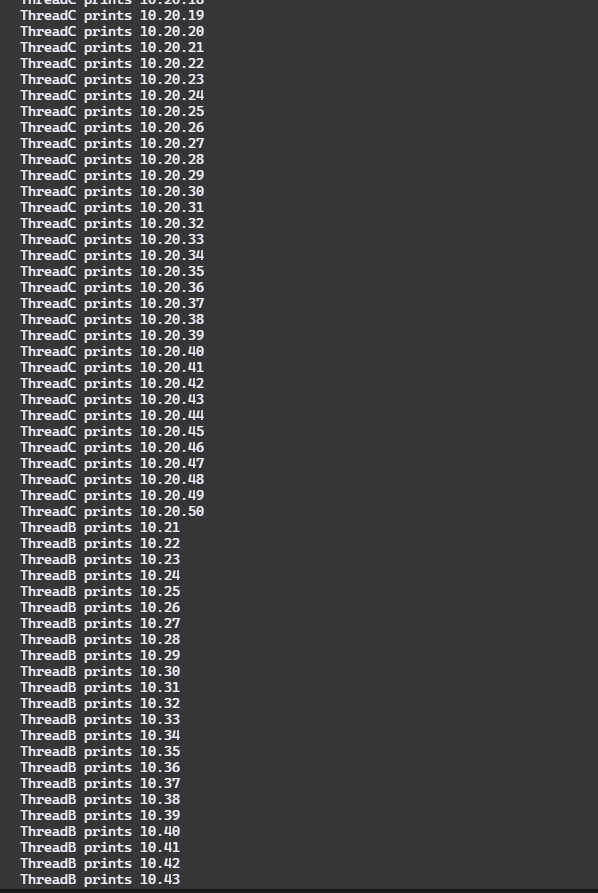
class ResourceLock{

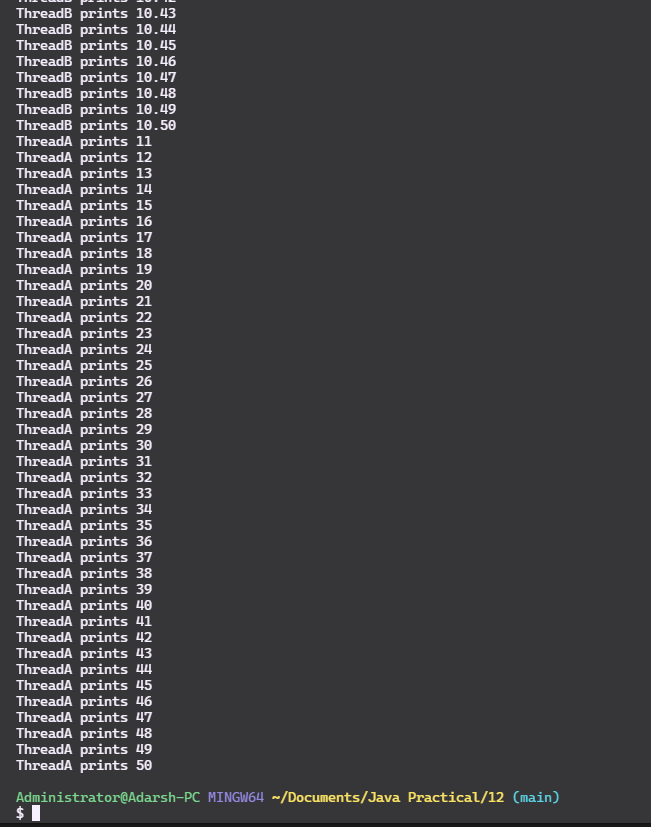
public volatile int flag = 1;

}

**Output – 12**







**Problem 13: There are 50 strings in a shared collection. Write a Java program to create two threads which are deleting a line from the collection at a time. All threads should remain alive until the collection is not empty.**

// Java program to Illustrate notify() method in Thread

// Synchronization.

// Importing required classes

import java.io.\*;

import java.util.\*;

// Class 1

// Thread1

// Helper class extending Thread class

public class SharedCollection {

// Main driver method

public static void main(String[] args)

{

SharedResource resource = new SharedResource();

// Creating object(thread) of class 2

ThreadA obj1 = new ThreadA(resource);

ThreadB obj2 = new ThreadB(resource);

System.out.println("\n\* Removing all " + resource.list.size() + " Countries from the list...");

// Starting the thread

obj1.start();

obj2.start();

}

}

// Class 2

// Thread2

// Helper class extending Thread class

class ThreadA extends Thread{

SharedResource resource;

ThreadA(SharedResource resource){

this.resource = resource;

}

@Override

public void run() {

try{

synchronized (resource) {

while (!resource.list.isEmpty()) {

while(resource.flag !=1 ){

resource.wait();

}

if(!resource.list.isEmpty()) {

String item = resource.list.get(0);

resource.list.remove(0);

System.out.println("x - ThreadA removed "+ item + " ("+ resource.list.size() + " items remaining.)");

}

resource.flag = 2;

resource.notify();

}

}

}catch (Exception e) {

System.out.println("Exception 1 :"+e.getMessage());

}

}

}

class ThreadB extends Thread{

SharedResource resource;

ThreadB(SharedResource resource){

this.resource = resource;

}

@Override

public void run() {

try{

synchronized (resource) {

while (!resource.list.isEmpty()) {

while(resource.flag != 2){

resource.wait();

}

if(!resource.list.isEmpty()) {

String item = resource.list.get(0);

resource.list.remove(0);

System.out.println("x - ThreadB removed "+ item + " ("+ resource.list.size() + " items remaining.)");

}

resource.flag = 1;

resource.notifyAll();

}

}

}catch (Exception e) {

System.out.println("Exception 2 :"+e.getMessage());

}

}

}

class SharedResource{

public volatile int flag = 1;

public List <String> list = Collections.synchronizedList(

new ArrayList<String>(

(Arrays.asList(

"Afghanistan",

"Australia",

"Bulgaria",

"China",

"Denmark",

"Egypt",

"Finland",

"France",

"Germany",

"Iceland",

"India",

"Indonesia",

"Iraq",

"Ireland",

"Israel",

"Italy",

"Jamaica",

"Japan",

"Kazakhstan",

"Kuwait",

"Nauru",

"Nepal",

"Netherlands",

"New Zealand",

"Norway",

"Pakistan",

"Philippines",

"Poland",

"Portugal",

"Qatar",

"Romania",

"Russia",

"Saudi Arabia",

"Serbia",

"Singapore",

"Spain",

"Sri Lanka",

"Switzerland",

"Syria",

"Tajikistan",

"Tanzania",

"Thailand",

"Tunisia",

"Turkey",

"Uganda",

"Ukraine",

"United Kingdom",

"Vietnam",

"Yemen",

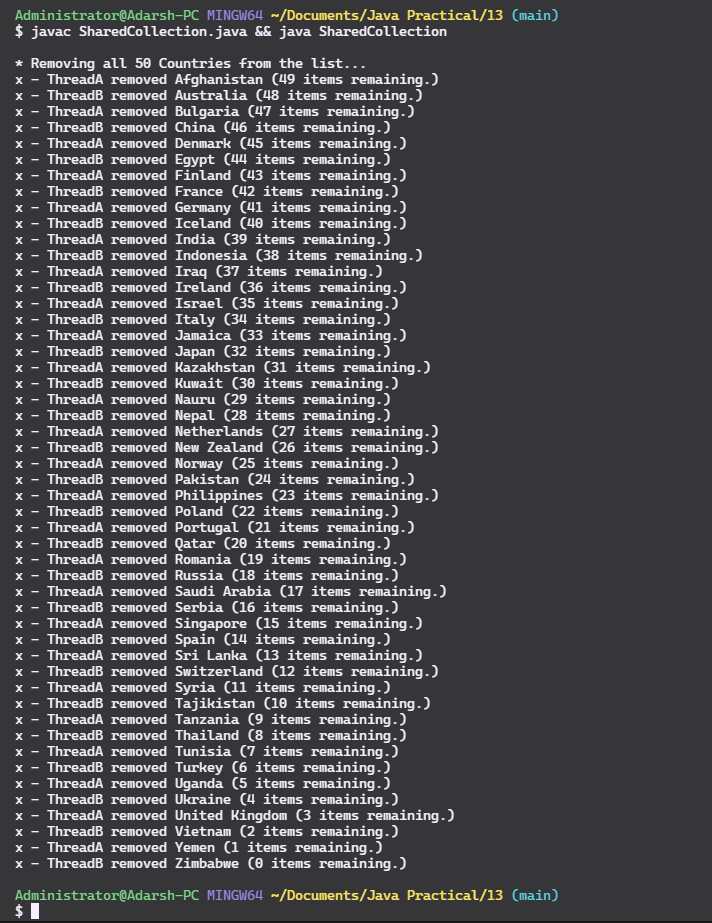
"Zimbabwe"

)

)));

}

**Output -13**



**Problem 14: Write a Java program to create two threads. First thread appends a line of text into a file at a time and is alive till 5 lines are appended. Second thread displays the lines from the file at a time and goes off if all 5 lines are displayed at a time.**

import java.io.\*;

import java.util.\*;

public class FileAppend {

// Main driver method

public static void main(String[] args)

{

SharedResource resource = new SharedResource();

ThreadA obj1 = new ThreadA(resource);

ThreadB obj2 = new ThreadB(resource);

// Starting the thread

obj1.start();

obj2.start();

}

}

class ThreadA extends Thread{

SharedResource resource;

ThreadA(SharedResource resource){

this.resource = resource;

}

@Override

public void run() {

try{

synchronized (resource) {

for (String quote : resource.list) {

while(resource.flag !=1 ){

resource.wait();

}

// File writer, append? true

FileWriter fw = new FileWriter("quotes.txt",true);

fw.write(quote + "\n");

// need to close to save the changes for other thread!

fw.close();

resource.flag = 2;

resource.notify();

}

}

}catch (Exception e) {

System.out.println("Exception 1 :"+e.getMessage());

}

}

}

class ThreadB extends Thread{

SharedResource resource;

ThreadB(SharedResource resource){

this.resource = resource;

}

@Override

public void run() {

try{

synchronized (resource) {

// resource list size not required, it will be five always

for (int i = 0; i< resource.list.size(); i++) {

while(resource.flag != 2){

resource.wait();

}

String line;

try {

BufferedReader br = new BufferedReader(new FileReader("quotes.txt"));

for (int j= 0; j < i; j++)

br.readLine();

line = br.readLine();

System.out.println(line);

}

catch(IOException e){

System.out.println(e);

}

resource.flag = 1;

resource.notifyAll();

}

}

}catch (Exception e) {

System.out.println("Exception 2 :"+e.getMessage());

}

}

}

class SharedResource{

public volatile int flag = 1;

public List <String> list = Collections.synchronizedList(

new ArrayList<String>(

(Arrays.asList(

"The way to get started is to quit talking and begin doing. -Walt Disney",

"Life is what happens when you're busy making other plans. -John Lennon",

"It is during our darkest moments that we must focus to see the light. -Aristotle",

"Whoever is happy will make others happy too. -Anne Frank",

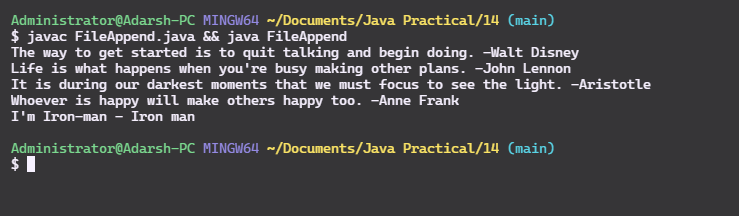
"I'm Iron-man - Iron man"

)

)));

}

**Output – 14**



**Problem 15: Write a JDBC program to connect to a database and demonstrate the Statement.**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

class Driver{

public static void main(String args[]){

Connection conn = null;

String url = "jdbc:mysql://localhost:3306/java\_practicals";

Properties property = new Properties();

property.put("user", "root");

property.put("password", "superdoge1234");

try{

conn = DriverManager.getConnection(url,property);

System.out.println("Connected to local mysql instance");

// Create the statement object and sql string.

Statement statement = conn.createStatement();

String sql = "select \* from employees";

// Execute the query

ResultSet result = statement.executeQuery(sql);

while (result.next()) {

String id = result.getString("id");

String name = result.getString("name");

String age = result.getString("age");

String address = result.getString("city");

String salary = result.getString("salary");

System.out.println(id+" - " + " Name: "+ name + ", Age: "+age+", address:"+ address+", Salary: "+salary);

}

}catch(SQLException e){

e.printStackTrace();

}finally{

System.out.println("Closing SQL connection.");

try{

conn.close();

}catch(SQLException e){

e.printStackTrace();

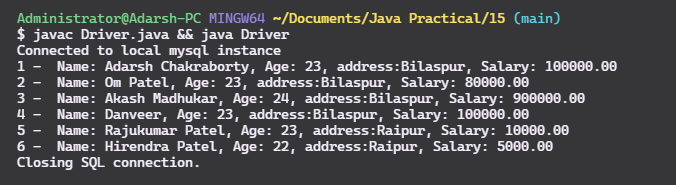
}

}

}

}

**Output – 15**



**Problem 16: Write a JDBC program to connect to a database and demonstrate the PreparedStatement.**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

import java.util.Scanner;

class Driver{

public static void main(String args[]){

Connection conn = null;

String url = "jdbc:mysql://localhost:3306/java\_practicals";

Properties property = new Properties();

property.put("user", "root");

property.put("password", "superdoge1234");

try{

Scanner sc = new Scanner(System.in);

conn = DriverManager.getConnection(url,property);

System.out.println("Connected to local mysql instance");

// Get the user input

System.out.println("Find employees based in city:\nEnter City:");

String cityQuery = sc.nextLine();

// Create the PreparedStatement object and pass sql string.

String sqlString = "SELECT \* FROM employees WHERE city like ?";

PreparedStatement statement = conn.prepareStatement(sqlString);

// Query index (?), value;

statement.setString(1, cityQuery);

// Execute the query

ResultSet result = statement.executeQuery();

System.out.println("Employees living in "+ cityQuery);

while (result.next()) {

String id = result.getString("id");

String name = result.getString("name");

String age = result.getString("age");

String address = result.getString("city");

String salary = result.getString("salary");

System.out.println(id+" - " + " Name: "+ name + ", Age: "+age+", address:"+ address+", Salary: "+salary);

}

}catch(SQLException e){

e.printStackTrace();

}finally{

System.out.println("Closing SQL connection.");

try{

conn.close();

}catch(SQLException e){

e.printStackTrace();

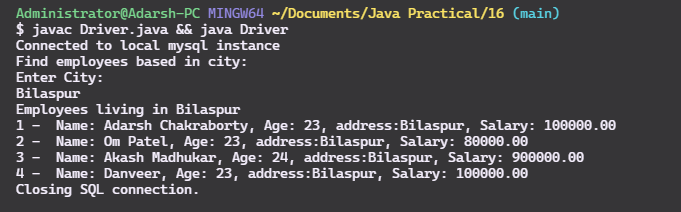
}

}

}

}

**Output – 16**



**Problem 17: Write a JDBC program to connect to a database and demonstrate the CallableStatement.**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.CallableStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

import java.util.Scanner;

class Driver{

public static void main(String args[]){

Connection conn = null;

String url = "jdbc:mysql://localhost:3306/java\_practicals";

Properties property = new Properties();

property.put("user", "root");

property.put("password", "superdoge1234");

try{

conn = DriverManager.getConnection(url,property);

System.out.println("Connected to local mysql instance");

// Create the PreparedStatement object and pass sql string.

String sqlString = "{ call getTotalEmployees(?) }";

CallableStatement statement = conn.prepareCall(sqlString);

// Register the out parameter and execute update query.

statement.registerOutParameter(1, java.sql.Types.INTEGER);

statement.executeUpdate();

// Getting the outout parameter from the callableStatement object.

int totalEmployees = statement.getInt(1);

System.out.println("Total Employees: "+ totalEmployees);

}catch(SQLException e){

e.printStackTrace();

}finally{

System.out.println("Closing SQL connection.");

try{

conn.close();

}catch(SQLException e){

e.printStackTrace();

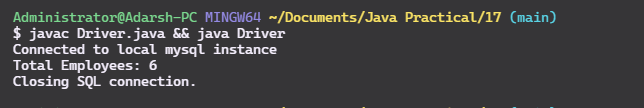
}

}

}

}

**Output – 17**



**Problem 18: Write a Servlet program to connect to a database and generate the page contains Tables combo, Fields combo and data area:**

**Table combo contains all tables of the database.**

**On selection of table combo Field combo contains all fields of selected table.**

**On selection of field combo all data of selected field are to be displayed on the data area.**

**If we select all fields from Field combo then respective table data are to be displayed.**

// GenerateIndexPage.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.DatabaseMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class GenerateIndexPage extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

Connection conn = null;

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

// Normal page generation

pw.println("""

<!DOCTYPE html>

<html lang=\"en\">

<head>

<meta charset=\"UTF-8\" />

<meta http-equiv=\"X-UA-Compatible\" content=\"IE=edge\" />

<meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\" />

<title>DB TABLE</title>

<link rel=\"stylesheet\" href=\"style.css\" />

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

.text-Bold {

font-size: medium;

font-weight: bold;

}

.data-area {

margin-top: 1.5rem;

display: flex;

flex-direction: column;

align-items: center;

margin: auto;

}

.data-area table {

display: block;

margin-top: 1rem;

}

.container {

width: 70%;

margin: auto;

display: flex;

flex-direction: column;

gap: 1rem;

align-items: center;

margin-top: 4.5rem;

}

.select\_style {

overflow: hidden;

}

.select\_style select {

-webkit-appearance: none;

appearance: none;

width: 120%;

background: none;

background: transparent;

border: none;

outline: none;

}

.select\_style {

background: #fff;

overflow: hidden;

display: inline-block;

color: #525252;

font-weight: 300;

-webkit-border-radius: 5px 4px 4px 5px/5px 5px 4px 4px;

-moz-border-radius: 5px 4px 4px 5px/5px 5px 4px 4px;

border-radius: 5px 4px 4px 5px/5px 5px 4px 4px;

-webkit-box-shadow: 0 0 5px rgba(123, 123, 123, 0.2);

-moz-box-shadow: 0 0 5px rgba(123, 123, 123, 0.2);

box-shadow: 0 0 5px rgba(123, 123, 123, 0.2);

border: solid 1px #dadada;

font-family: 'helvetica neue', arial;

position: relative;

cursor: pointer;

padding: 10px 15px;

}

.select\_style span {

position: absolute;

right: 10px;

width: 10px;

height: 10px;

background: url('http://projects.authenticstyle.co.uk/niceselect/arrow.png')

no-repeat;

top: 50%;

margin-top: -4px;

}

#dataTable {

font-family: Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

#dataTable td,

#dataTable th {

border: 1px solid #ddd;

padding: 8px;

}

#dataTable tr:nth-child(even) {

background-color: #f2f2f2;

}

#dataTable tr:hover {

background-color: #ddd;

}

#dataTable th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: #546e7a;

color: white;

}

</style>

</head>

<body>

<div class=\"container\">

<div class=\"\">

<span class=\"text-Bold\">Select Table: &nbsp;</span>

<select name=\"table\_name\" id=\"table\_name\" class=\"select\_style\">

<option value=\"\" disabled selected>Select Table</option>

""");

try{

String url = "jdbc:mysql://localhost:3306/java\_practicals";

Properties property = new Properties();

property.put("user", "root");

property.put("password", "superdoge1234");

Class.forName("com.mysql.cj.jdbc.Driver");

conn = DriverManager.getConnection(url,property);

DatabaseMetaData metaData = conn.getMetaData();

String[] types = {"TABLE"};

ResultSet tables = metaData.getTables("java\_practicals", null, "%", types);

while (tables.next()) {

String tableName = tables.getString("TABLE\_NAME");

pw.println("<option value=\""+tableName+"\">"+tableName+"</option>");

}

pw.println("""

</select>

</div>

<div class=\"\">

<span class=\"text-Bold\">Select Field: &nbsp;</span>

<select name=\"field\_name\" id=\"field\_name\" class=\"select\_style\">

<option value=\"\" disabled selected>Select Field</option>

</select>

</div>

<div class=\"data-area\"></div>

</div>

</body>

<script>

console.log('Script attached');

const baseUrl = window.location.href;

const tableSelector = document.querySelector('#table\_name');

const fieldSelector = document.querySelector('#field\_name');

const dataArea = document.querySelector('.data-area');

tableSelector.addEventListener('change', tableChangeHandler);

fieldSelector.addEventListener('change', fieldChangeHandler);

let currentOptions = [];

async function tableChangeHandler(e) {

currentOptions = [];

dataArea.innerHTML = '';

const req = await fetch(baseUrl + `/GetTableFields?table=${e.target.value}`);

const data = await req.text();

removeAll(fieldSelector);

const fields = data.split('\\r\\n');

const newOption = document.createElement('option');

const optionText = document.createTextNode('All');

newOption.appendChild(optionText);

newOption.setAttribute('value', 'all');

fieldSelector.appendChild(newOption);

fields.forEach((field) => {

if (field.trim()) {

const newOption = document.createElement('option');

const optionText = document.createTextNode(field);

newOption.appendChild(optionText);

newOption.setAttribute('value', field);

fieldSelector.appendChild(newOption);

currentOptions.push(field);

}

});

}

async function fieldChangeHandler(e) {

const selectedTable = tableSelector.options[tableSelector.selectedIndex];

const selectedField = fieldSelector.options[fieldSelector.selectedIndex];

const req = await fetch(

baseUrl +

`/GetTableData?table=${selectedTable.value}&field=${selectedField.value}`

);

const data = await req.text();

renderDataTable(selectedTable.value, selectedField.value, data);

}

function removeAll(selectBox) {

while (selectBox.options.length > 0) {

selectBox.remove(0);

}

}

function renderDataTable(tableName, fieldName, tableContent) {

if (fieldName.toLowerCase() === 'all') {

let htmlString = `<h2>${tableName}<h2><table id='dataTable'><tr>`;

for (let field of currentOptions) {

htmlString += `<th>${field}</th>`;

}

htmlString += `</tr >`;

// Array of fields

const records = tableContent.split('$$'); // Record split

for (let record of records) {

if (record) {

const temp = record.split('\\r\\n');

htmlString += `</tr>`;

for (let temp2 of temp) {

if (temp2) {

htmlString += `

<td>${temp2.trim()}</td>

`;

}

}

htmlString += `</tr>`;

}

}

dataArea.innerHTML = htmlString + '</table>';

return;

}

const data = tableContent.split('$$');

let htmlString = `<h2>${tableName}<h2><table id='dataTable'><tr>

<th>${fieldName}</th>

</tr>`;

for (let item of data) {

if (item.trim()) {

htmlString += `<tr>

<td>${item.trim()}</td>

</tr>`;

}

}

dataArea.innerHTML = htmlString + '</table>';

}

</script>

</html>

""");

}catch(Exception e){

e.printStackTrace();

}finally{

try{

conn.close();

}catch(SQLException e){

e.printStackTrace();

}

}

}}

// GetTableData.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class GetTableData extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

String tablName = req.getParameter("table");

String fieldName = req.getParameter("field");

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

if((tablName != null && tablName.trim() != "") && (fieldName != null && fieldName.trim() != "") ){

Connection conn = null;

try{

String field = fieldName;

if(fieldName.toLowerCase().equals("all")){

field = "\*";

}

String url = "jdbc:mysql://localhost:3306/java\_practicals";

Properties property = new Properties();

property.put("user", "root");

property.put("password", "superdoge1234");

Class.forName("com.mysql.cj.jdbc.Driver");

conn = DriverManager.getConnection(url,property);

Statement statement = conn.createStatement();

String sql = "select "+field+" from "+tablName;

ResultSet rs = statement.executeQuery(sql);

ResultSetMetaData rsMetaData = rs.getMetaData();

int count = rsMetaData.getColumnCount();

while (rs.next()) {

for(int i = 1 ; i <= count; i++){

pw.println(rs.getString(i) + " ");

}

pw.println("$$");

}

}catch(Exception e){

e.printStackTrace();

pw.println("Error: Something went wrong, try again later...");

}finally{

try{

conn.close();

}catch(SQLException e){

e.printStackTrace();

}

}

}else {

pw.println("Error: Table and field names are required!");

}

}}

// GetTableFields.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class GetTableFields extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

String tablName = req.getParameter("table");

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

if(tablName != null && tablName.trim() != ""){

Connection conn = null;

try{

String url = "jdbc:mysql://localhost:3306/java\_practicals";

Properties property = new Properties();

property.put("user", "root");

property.put("password", "superdoge1234");

Class.forName("com.mysql.cj.jdbc.Driver");

conn = DriverManager.getConnection(url,property);

Statement statement = conn.createStatement();

String sql = "select \* from "+tablName;

ResultSet rs = statement.executeQuery(sql);

ResultSetMetaData rsMetaData = rs.getMetaData();

int count = rsMetaData.getColumnCount();

for(int i = 1; i<=count; i++) {

pw.println(rsMetaData.getColumnName(i));

}

}catch(Exception e){

e.printStackTrace();

pw.println("Error: Something went wrong, try again later...");

}finally{

try{

conn.close();

}catch(SQLException e){

e.printStackTrace();

}

}

}else {

pw.println("Error: Table name is required!");

}

}}

// web.xml file

<web-app>

<servlet>

<servlet-name>getTableData</servlet-name>

<servlet-class>GetTableData</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>getTableData</servlet-name>

<url-pattern>/GetTableData</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>getTableFields</servlet-name>

<servlet-class>GetTableFields</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>getTableFields</servlet-name>

<url-pattern>/GetTableFields</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>adarshchakraborty</servlet-name>

<servlet-class>GenerateIndexPage</servlet-class>

</servlet>

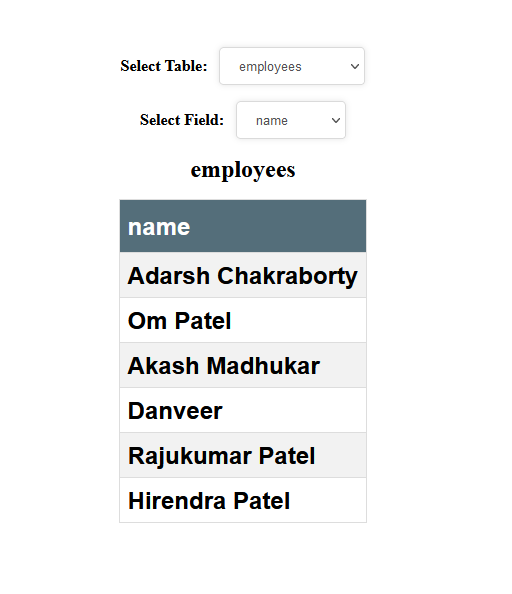
<servlet-mapping>

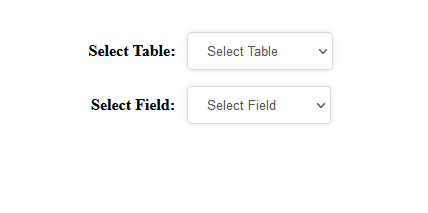
<servlet-name>adarshchakraborty</servlet-name>

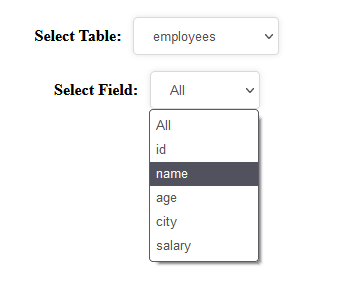
<url-pattern>/</url-pattern>

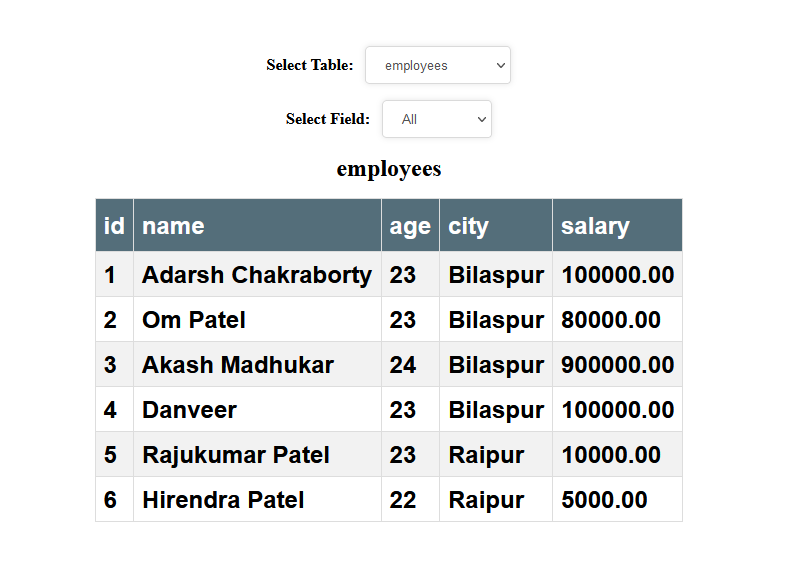
</servlet-mapping>

</web-app>

**Output – 18**









**19. Implement Session tracking in servlet using:**

**a. Hidden from field**

**b. Url re-write**

**c. Cookies**

**d. Session**

// Servlet1Cookie.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet1Cookie extends HttpServlet{

public void doPost(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

String firstname = req.getParameter("firstname");

String lastname = req.getParameter("lastname");

String email = req.getParameter("email");

String country = req.getParameter("country");

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

Cookie cfirstname = new Cookie("firstname", firstname);

Cookie clastname = new Cookie("lastname", lastname);

Cookie cmail = new Cookie("email", email);

Cookie ccountry = new Cookie("country", country);

res.addCookie(cfirstname);

res.addCookie(clastname);

res.addCookie(cmail);

res.addCookie(ccountry);

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstname);

pw.println("""

!</h2>

<p>You've been logged in.</p>

<div class=\"container\">

<a href=\"serv2cookie\">View full profile on servlet2...</a>

</div>

</body>

</html>

""");

}

}

// Servlet1HiddenForm.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet1HiddenForm extends HttpServlet{

public void doPost(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

String firstname = req.getParameter("firstname");

String lastname = req.getParameter("lastname");

String email = req.getParameter("email");

String country = req.getParameter("country");

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

String htmlForm = "<form action='serv2form' method='post'>"

+ "<input type='hidden' name='firstname' value='"+firstname+"' />"

+ "<input type='hidden' name='lastname' value='"+lastname+"' />"

+ "<input type='hidden' name='email' value='"+email+"' />"

+ "<input type='hidden' name='country' value='"+country+"' />"

+ "<button type='submit'>View profile on servlet2</button>"

+ "</form>";

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstname);

pw.println("""

!</h2>

<p>You've been logged in.</p>

<div class=\"container\">

""");

pw.println(htmlForm);

pw.println("""

</div>

</body>

</html>

""");

}

}

// Servlet1HttpSession.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet1HttpSession extends HttpServlet{

public void doPost(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

String firstname = req.getParameter("firstname");

String lastname = req.getParameter("lastname");

String email = req.getParameter("email");

String country = req.getParameter("country");

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

HttpSession session= req.getSession();

session.setAttribute("firstname",firstname);

session.setAttribute("lastname",lastname);

session.setAttribute("email",email);

session.setAttribute("country",country);

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstname);

pw.println("""

!</h2>

<p>You've been logged in.</p>

<div class=\"container\">

<a href='serv2session'>View full profile on Servlet2.</a>

</div>

</body>

</html>

""");

}

}

// Servlet1UrlRewrite.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet1UrlRewrite extends HttpServlet{

public void doPost(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

String firstname = req.getParameter("firstname");

String lastname = req.getParameter("lastname");

String email = req.getParameter("email");

String country = req.getParameter("country");

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

String customURL = "?firstname="+firstname+"&lastname="+lastname+"&email="+email+"&country="+country;

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstname);

pw.println("""

!</h2>

<p>You've been logged in.</p>

<div class=\"container\">

<a href=\"serv2url""");

pw.println(customURL);

pw.println("""

\">View full profile on servlet2...</a>

</div>

</body>

</html>

""");

}

}

// Servlet2Cookie.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet2Cookie extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

Cookie[] cookies = req.getCookies();

String firstName = "";

String lastName = "";

String email = "";

String country = "";

if(cookies == null){

pw.println("You are not logged in!");

pw.println("<a href=\"cookies.html\">Go back to login page.</a>");

}else{

for(Cookie c : cookies){

String cookieHeader = c.getName();

if(cookieHeader.equals("firstname")){

firstName = c.getValue();

}

if(cookieHeader.equals("lastname")){

lastName = c.getValue();

}

if(cookieHeader.equals("email")){

email = c.getValue();

}

if(cookieHeader.equals("country")){

country = c.getValue();

}

}

}

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstName);

pw.println("""

!</h2>

<p>Thanks for registering.. We have received your following details:</p>

<div class=\"container\">

""");

pw.println("<h2>Name: "+firstName+ " "+ lastName + "</h2>");

pw.println("<h2>Country: "+country+ "</h2>");

pw.println("<h2>Email: "+email+ "</h2>");

pw.println("""

<p>

We will contact you through your registered email address:

<a href=\"mailto:

""");

pw.println(email+"\">");

pw.println(email);

pw.println("""

</a>

</p>

</div>

</body>

</html>

""");

}

}

// Servlet2HiddenForm.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet2HiddenForm extends HttpServlet{

public void doPost(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

String firstName = req.getParameter("firstname");

String lastName = req.getParameter("lastname");

String email = req.getParameter("email");

String country = req.getParameter("country");

if(firstName != null && lastName != null && email != null && country != null){

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstName);

pw.println("""

!</h2>

<p>Thanks for registering.. We have received your following details:</p>

<div class=\"container\">

""");

pw.println("<h2>Name: "+firstName+ " "+ lastName + "</h2>");

pw.println("<h2>Country: "+country+ "</h2>");

pw.println("<h2>Email: "+email+ "</h2>");

pw.println("""

<p>

We will contact you through your registered email address:

<a href=\"mailto:

""");

pw.println(email+"\">");

pw.println(email);

pw.println("""

</a>

</p>

</div>

</body>

</html>

""");

}else{

pw.println("You are not logged in!");

pw.println("<a href=\"url.html\">Go back to login page.</a>");

}

}

}

// Servlet2HttpSession.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet2HttpSession extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

String firstName = "";

String lastName = "";

String email = "";

String country = "";

HttpSession session=req.getSession(false);

if(session != null){

firstName = (String) session.getAttribute("firstname");

lastName = (String) session.getAttribute("lastname");

email = (String) session.getAttribute("email");

country = (String) session.getAttribute("country");

if(firstName != "" && lastName != "" && email != "" && country != ""){

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstName);

pw.println("""

!</h2>

<p>Thanks for registering.. We have received your following details:</p>

<div class=\"container\">

""");

pw.println("<h2>Name: "+firstName+ " "+ lastName + "</h2>");

pw.println("<h2>Country: "+country+ "</h2>");

pw.println("<h2>Email: "+email+ "</h2>");

pw.println("""

<p>

We will contact you through your registered email address:

<a href=\"mailto:

""");

pw.println(email+"\">");

pw.println(email);

pw.println("""

</a>

</p>

</div>

</body>

</html>

""");

}else{

pw.println("You are not logged in!");

pw.println("<a href=\"url.html\">Go back to login page.</a>");

}

}else {

pw.println("You are not logged in!");

pw.println("<a href=\"session.html\">Go back to login page.</a>");

}

}

}

// Servlet2UrlRewrite.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class Servlet2UrlRewrite extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

String firstName = req.getParameter("firstname");

String lastName = req.getParameter("lastname");

String email = req.getParameter("email");

String country = req.getParameter("country");

if(firstName != null && lastName != null && email != null && country != null){

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstName);

pw.println("""

!</h2>

<p>Thanks for registering.. We have received your following details:</p>

<div class=\"container\">

""");

pw.println("<h2>Name: "+firstName+ " "+ lastName + "</h2>");

pw.println("<h2>Country: "+country+ "</h2>");

pw.println("<h2>Email: "+email+ "</h2>");

pw.println("""

<p>

We will contact you through your registered email address:

<a href=\"mailto:

""");

pw.println(email+"\">");

pw.println(email);

pw.println("""

</a>

</p>

</div>

</body>

</html>

""");

}else{

pw.println("You are not logged in!");

pw.println("<a href=\"url.html\">Go back to login page.</a>");

}

}

}

// web.xml

<web-app>

<!-- Cookie Method -->

<servlet>

<servlet-name>Servlet1Cookie</servlet-name>

<servlet-class>Servlet1Cookie</servlet-class>

</servlet>

<servlet>

<servlet-name>Servlet2Cookie</servlet-name>

<servlet-class>Servlet2Cookie</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet1Cookie</servlet-name>

<url-pattern>/serv1cookie</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet2Cookie</servlet-name>

<url-pattern>/serv2cookie</url-pattern>

</servlet-mapping>

<!-- URL Rewrite method -->

<servlet>

<servlet-name>Servlet1UrlRewrite</servlet-name>

<servlet-class>Servlet1UrlRewrite</servlet-class>

</servlet>

<servlet>

<servlet-name>Servlet2UrlRewrite</servlet-name>

<servlet-class>Servlet2UrlRewrite</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet2UrlRewrite</servlet-name>

<url-pattern>/serv2url</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet1UrlRewrite</servlet-name>

<url-pattern>/serv1url</url-pattern>

</servlet-mapping>

<!-- Hidden Form Field -->

<servlet>

<servlet-name>Servlet1HiddenForm</servlet-name>

<servlet-class>Servlet1HiddenForm</servlet-class>

</servlet>

<servlet>

<servlet-name>Servlet2HiddenForm</servlet-name>

<servlet-class>Servlet2HiddenForm</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet1HiddenForm</servlet-name>

<url-pattern>/serv1form</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet2HiddenForm</servlet-name>

<url-pattern>/serv2form</url-pattern>

</servlet-mapping>

<!-- Session -->

<servlet>

<servlet-name>Servlet1HttpSession</servlet-name>

<servlet-class>Servlet1HttpSession</servlet-class>

</servlet>

<servlet>

<servlet-name>Servlet2HttpSession</servlet-name>

<servlet-class>Servlet2HttpSession</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet1HttpSession</servlet-name>

<url-pattern>/serv1session</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet2HttpSession</servlet-name>

<url-pattern>/serv2session</url-pattern>

</servlet-mapping>

</web-app>

// cookies.html

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

input[type='text'],

select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type='submit'] {

width: 100%;

background-color: #4caf50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type='submit']:hover {

background-color: #45a049;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

</style>

<body>

<h2 class="heading">Session tracking demo</h2>

<p>Demonstration of Session tracking using cookies.</p>

<div class="container">

<form action="serv1cookie" method="post">

<label for="fname">First Name</label>

<input

type="text"

id="fname"

name="firstname"

placeholder="Your name.."

/>

<label for="lastname">Last Name</label>

<input

type="text"

id="lname"

name="lastname"

placeholder="Your last name.."

/>

<label for="email">E-mail</label>

<input

type="text"

id="lname"

name="email"

placeholder="Your email address.."

/>

<label for="country">Country</label>

<select id="country" name="country">

<option value="india">India</option>

<option value="pakistan">Pakistan</option>

<option value="russia">Russia</option>

</select>

<input type="submit" value="Submit" />

</form>

</div>

</body>

</html>

// form.html

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

input[type='text'],

select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type='submit'] {

width: 100%;

background-color: #4caf50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type='submit']:hover {

background-color: #45a049;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

</style>

<body>

<h2 class="heading">Session tracking demo</h2>

<p>Demonstration of Session tracking using hidden form field.</p>

<div class="container">

<form action="serv1form" method="post">

<label for="fname">First Name</label>

<input

type="text"

id="fname"

name="firstname"

placeholder="Your name.."

/>

<label for="lastname">Last Name</label>

<input

type="text"

id="lname"

name="lastname"

placeholder="Your last name.."

/>

<label for="email">E-mail</label>

<input

type="text"

id="lname"

name="email"

placeholder="Your email address.."

/>

<label for="country">Country</label>

<select id="country" name="country">

<option value="india">India</option>

<option value="pakistan">Pakistan</option>

<option value="russia">Russia</option>

</select>

<input type="submit" value="Submit" />

</form>

</div>

</body>

</html>

// index.html

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

display: flex;

flex-direction: column;

gap: 1rem;

padding: 2rem 0;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

a:visited {

color: blue;

}

</style>

<body>

<h2 class="heading">Session Tracking Demo</h2>

<p>Choose any of the tracking method to continue...</p>

<div class="container">

<a href="form.html">Hidden Form field</a>

<a href="url.html">URL Re-write</a>

<a href="cookies.html">Cookies</a>

<a href="session.html">Session</a>

</div>

</body>

</html>

// session.html

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

input[type='text'],

select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type='submit'] {

width: 100%;

background-color: #4caf50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type='submit']:hover {

background-color: #45a049;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

</style>

<body>

<h2 class="heading">Session tracking demo</h2>

<p>Demonstration of Session tracking using Session.</p>

<div class="container">

<form action="serv1session" method="post">

<label for="fname">First Name</label>

<input

type="text"

id="fname"

name="firstname"

placeholder="Your name.."

/>

<label for="lastname">Last Name</label>

<input

type="text"

id="lname"

name="lastname"

placeholder="Your last name.."

/>

<label for="email">E-mail</label>

<input

type="text"

id="lname"

name="email"

placeholder="Your email address.."

/>

<label for="country">Country</label>

<select id="country" name="country">

<option value="india">India</option>

<option value="pakistan">Pakistan</option>

<option value="russia">Russia</option>

</select>

<input type="submit" value="Submit" />

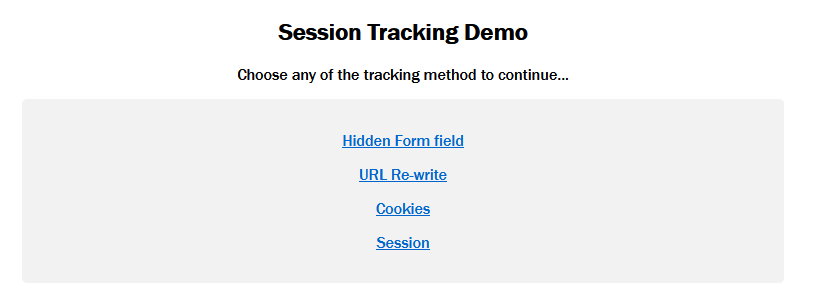
</form>

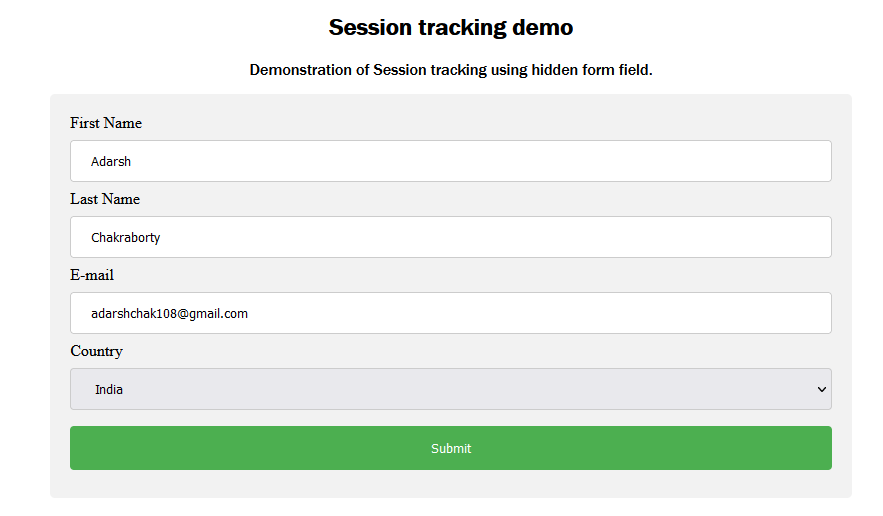
</div>

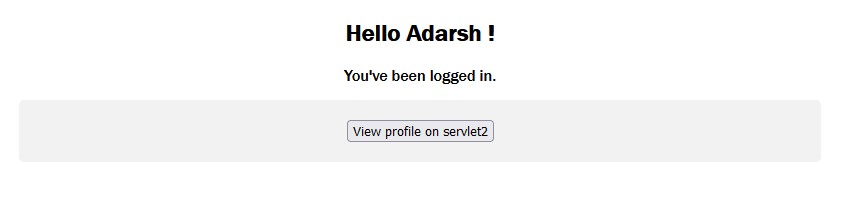
</body>

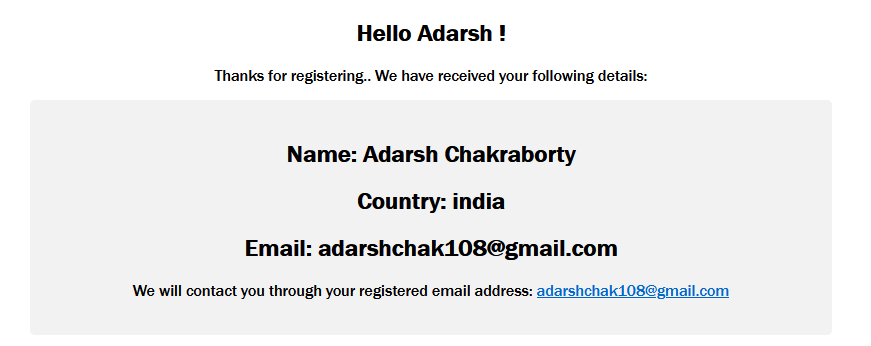
</html>

**Output – 19**









**Problem 20: Write a Servlet program to demonstrate initial parameter access from a servlet page.**

// GenerateIndexFile.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class GenerateIndexFile extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

pw.println("""

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Init parameter access</title>

<style>

.container {

width: 70%;

margin: auto;

text-align: center;

}

footer {

text-align: center;

}

</style>

</head>

<body>

<div class="container">

<h2>Welcome to home page.</h2>

<h3>Developer name is set as initial parameter on this servlet.</h3>

<hr />

</div>

<footer>

<p>This site is designed by <strong>

""");

ServletConfig config = getServletConfig();

String developer = config.getInitParameter("Developer");

pw.println(developer);

pw.println("""

</strong></p>

</footer>

</body>

</html>

""");

}

}

// web. Xml

<web-app>

<servlet>

<servlet-name>getIndexPage</servlet-name>

<servlet-class>GenerateIndexFile</servlet-class>

<init-param>

<param-name>Developer</param-name>

<param-value>Adarsh Chakraborty</param-value>

</init-param>

</servlet>

<servlet-mapping>

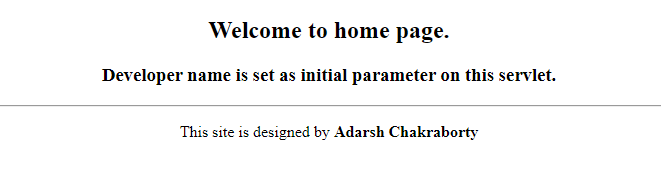
<servlet-name>getIndexPage</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**Output – 20**



**Problem 21: Write a Servlet program to demonstrate parameter access from a servlet page.**

// RegisterUser.java

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.DriverManager;

import java.sql.ResultSetMetaData;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Properties;

public class RegisterUser extends HttpServlet{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

String firstname = req.getParameter("firstname");

String lastname = req.getParameter("lastname");

String email = req.getParameter("email");

String country = req.getParameter("country");

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

pw.println("""

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

</style>

<body>

<h2 class=\"heading\">Hello """);

pw.println(firstname);

pw.println("""

!</h2>

<p>Thanks for registering.. We have received your following details:</p>

<div class=\"container\">

""");

pw.println("<h2>Name: "+firstname+ " "+ lastname + "</h2>");

pw.println("<h2>Country: "+country+ "</h2>");

pw.println("<h2>Email: "+email+ "</h2>");

pw.println("""

<p>

You will be contacted on your email address at:

<a href=\"mailto:

""");

pw.println(email+"\">");

pw.println(email);

pw.println("""

</a>

</p>

</div>

</body>

</html>

""");

}

}

// web.xml

<web-app>

<servlet>

<servlet-name>RegisterUser</servlet-name>

<servlet-class>RegisterUser</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>RegisterUser</servlet-name>

<url-pattern>/register</url-pattern>

</servlet-mapping>

</web-app>

// index.html

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

input[type='text'],

select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type='submit'] {

width: 100%;

background-color: #4caf50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type='submit']:hover {

background-color: #45a049;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

</style>

<body>

<h2 class="heading">Send your information to servlet</h2>

<p>

Servlet will access these information by accessing the request parameters.

</p>

<div class="container">

<form action="register">

<label for="fname">First Name</label>

<input

type="text"

id="fname"

name="firstname"

placeholder="Your name.."

/>

<label for="lastname">Last Name</label>

<input

type="text"

id="lname"

name="lastname"

placeholder="Your last name.."

/>

<label for="email">E-mail</label>

<input

type="text"

id="lname"

name="email"

placeholder="Your email address.."

/>

<label for="country">Country</label>

<select id="country" name="country">

<option value="india">India</option>

<option value="pakistan">Pakistan</option>

<option value="russia">Russia</option>

</select>

<input type="submit" value="Submit" />

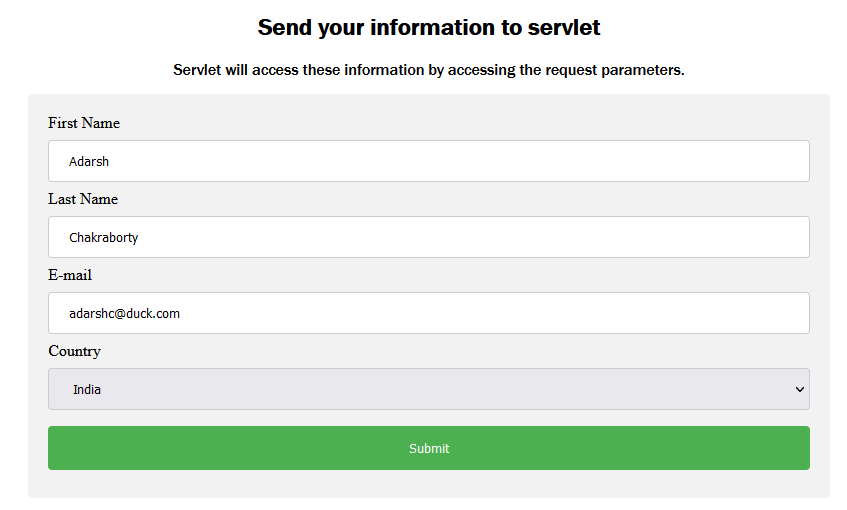
</form>

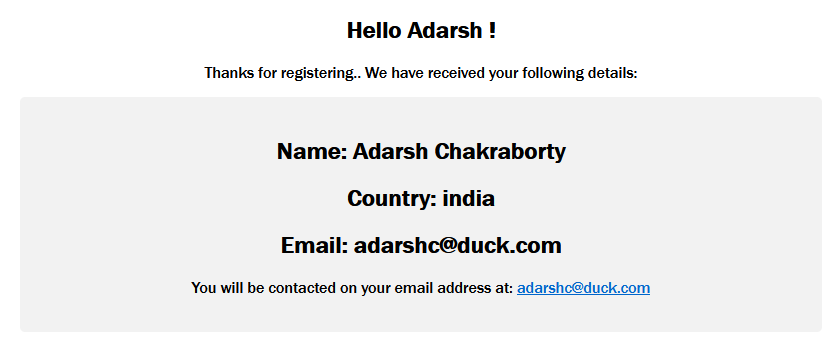
</div>

</body>

</html>

**Output – 21**





**Problem 22: Write a JSP program to demonstrate access bean object in different scopes. How to access bean object which was opened in a different page?**

// User.java (bean)

package p1;

public class User{

private String firstName, lastName, country, email;; //property

public void setFirstName(final String firstname){

firstName = firstname;

}

public String getFirstName(){

return firstName;

}

//

public void setLastName(final String lastname){

lastName = lastname;

}

public String getLastName(){

return lastName;

}

//

public void setCountry(final String usrCountry){

country = usrCountry;

}

public String getCountry(){

return country;

}

//

public void setEmail(final String userEmail){

email = userEmail;

}

public String getEmail(){

return email;

}

public String getFullName(){

return firstName + " " + lastName;

}

public String getBIO(){

return "Hello, I am "+ firstName+" " +lastName+ " and I am from "+ country + ".";

}

public String getMailtoUrl(){

return "<a href=mailto:"+email+">"+email+"</a>";

}

}

// index.html

<!DOCTYPE html>

<html>

<style>

html {

color: #212121;

}

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

input[type='text'],

select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #00695c;

border-radius: 4px;

box-sizing: border-box;

}

input[type='submit'] {

width: 100%;

background-color: #00695c;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type='submit']:hover {

background-color: #00695c;

}

div {

border-radius: 2px;

background-color: #b2dfdb;

padding: 20px;

box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;

}

</style>

<body>

<h2 class="heading">Send your information to servlet</h2>

<p>Servlet will access these information by Bean and JSP actions.</p>

<div class="container">

<form action="login.jsp" method="post">

<label for="fname">First Name</label>

<input

type="text"

id="fname"

name="firstName"

placeholder="Your name.."

/>

<label for="lastname">Last Name</label>

<input

type="text"

id="lname"

name="lastName"

placeholder="Your last name.."

/>

<label for="email">E-mail</label>

<input

type="text"

id="lname"

name="email"

placeholder="Your email address.."

/>

<label for="country">Country</label>

<select id="country" name="country">

<option value="india">India</option>

<option value="pakistan">Pakistan</option>

<option value="russia">Russia</option>

</select>

<input type="submit" value="Submit" />

</form>

</div>

</body>

</html>

// login.jsp

<jsp:useBean id="currentUser" class="p1.User" scope="application">

</jsp:useBean>

<jsp:setProperty name="currentUser" property="\*" />

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

color: #212121;

border-radius: 5px;

background-color: #80cbc4;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

box-shadow: rgba(0, 0, 0, 0.25) 0px 54px 55px,

rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px,

rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

}

</style>

<body>

<br />

<br />

<br />

<div class="container">

<h2 class="heading">

Hello <jsp:getProperty name="currentUser" property="firstName" />!

</h2>

<p>

You've been logged in with following email address: <br /><br />

<%= currentUser.getMailtoUrl() %>

</p>

<a href="profile.jsp">Click here to visit your profile page...</a>

</div>

</body>

</html>

// profile.jsp

<jsp:useBean id="currentUser" class="p1.User" scope="application">

</jsp:useBean>

<!DOCTYPE html>

<html>

<style>

html {

color: #263238;

}

.heading {

text-decoration: underline;

}

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 2px;

background-color: #4db6ac;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

box-shadow: rgba(0, 0, 0, 0.25) 0px 54px 55px,

rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px,

rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

}

.bio {

font-weight: 500;

font-size: large;

}

</style>

<body>

<h1 class="heading">User Profile</h1>

<p><span class="bio">BIO:</span> <i><%= currentUser.getBIO() %></i></p>

<div class="container">

<h2>Name: <%= currentUser.getFullName() %></h2>

<h2>Email: <jsp:getProperty name="currentUser" property="email" /></h2>

<h2>

Country: <jsp:getProperty name="currentUser" property="country" />

</h2>

<p>

You've will further notifications on your email address at: <%=

currentUser.getMailtoUrl() %>

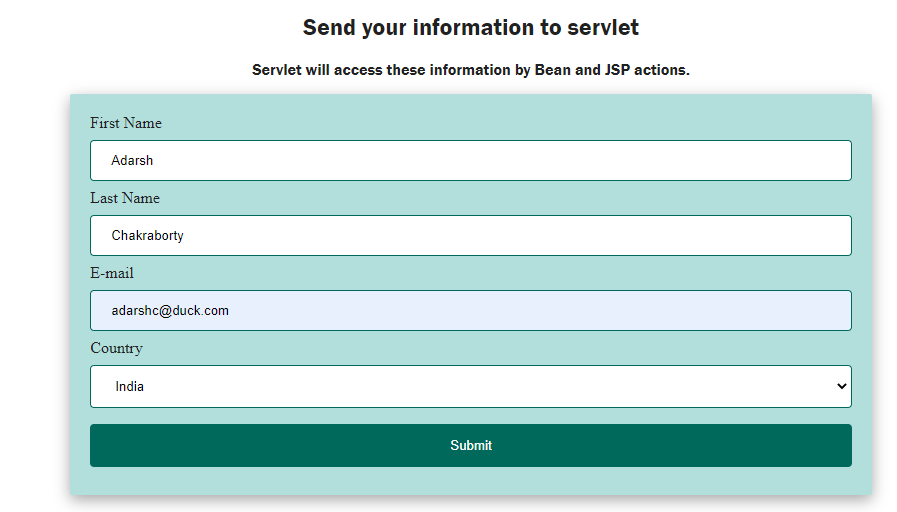
</p>

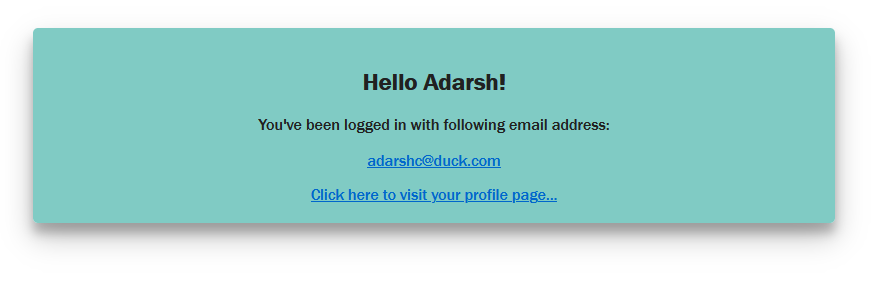
</div>

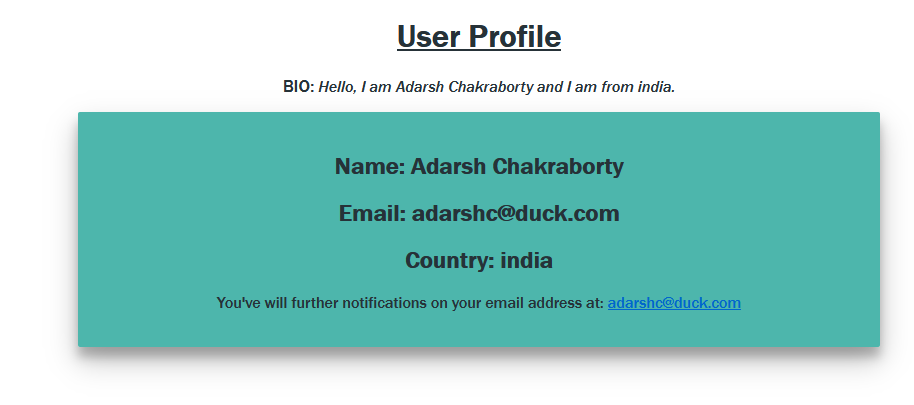
</body>

</html>

**Output – 22**







**Problem 23: Write a JSP program to demonstrate at least five action tags.**

// User.java (bean)

package p1;

public class User{

private String firstName, lastName, country, email;; //property

public void setFirstName(final String firstname){

firstName = firstname;

}

public String getFirstName(){

return firstName;

}

//

public void setLastName(final String lastname){

lastName = lastname;

}

public String getLastName(){

return lastName;

}

//

public void setCountry(final String usrCountry){

country = usrCountry;

}

public String getCountry(){

return country;

}

//

public void setEmail(final String userEmail){

email = userEmail;

}

public String getEmail(){

return email;

}

public String getFullName(){

return firstName + " " + lastName;

}

public String getBIO(){

return "Hello, I am "+ firstName+" " +lastName+ " and I am from "+ country + ".";

}

public String getMailtoUrl(){

return "<a href=mailto:"+email+">"+email+"</a>";

}

}

// index.html

<!DOCTYPE html>

<html>

<style>

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

input[type='text'],

select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type='submit'] {

width: 100%;

background-color: #0d47a1;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type='submit']:hover {

background-color: #3949ab;

}

div {

border-radius: 2px;

background-color: #e8eaf6;

padding: 20px;

box-shadow: rgba(0, 0, 0, 0.2) 0px 12px 28px 0px,

rgba(0, 0, 0, 0.1) 0px 2px 4px 0px,

rgba(255, 255, 255, 0.05) 0px 0px 0px 1px inset;

}

</style>

<body>

<h2 class="heading">Welcome!</h2>

<p>Enter all the details and click submit button.</p>

<div class="container">

<form action="login.jsp" method="post">

<label for="fname">First Name</label>

<input

type="text"

id="fname"

name="firstName"

placeholder="Your name.."

/>

<label for="lastname">Last Name</label>

<input

type="text"

id="lname"

name="lastName"

placeholder="Your last name.."

/>

<label for="email">E-mail</label>

<input

type="text"

id="lname"

name="email"

placeholder="Your email address.."

/>

<label for="country">Country</label>

<select id="country" name="country">

<option value="india">India</option>

<option value="pakistan">Pakistan</option>

<option value="russia">Russia</option>

</select>

<input type="submit" value="Submit" />

</form>

</div>

</body>

</html>

// indian.jsp

<jsp:useBean id="currentUser" class="p1.User" scope="application">

</jsp:useBean>

<!DOCTYPE html>

<html>

<style>

html {

background-color: #90caf9;

}

.heading {

text-decoration: underline;

}

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 2px;

background-color: #bbdefb;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

box-shadow: rgba(50, 50, 93, 0.25) 0px 13px 27px -5px,

rgba(0, 0, 0, 0.3) 0px 8px 16px -8px;

}

.bio {

font-weight: 500;

font-size: large;

}

</style>

<body>

<h1 class="heading">User Profile</h1>

<!-- 4. Jsp include action tag -->

<jsp:include page="userbio.jsp"></jsp:include>

<div class="container">

<h2>Name: <%= currentUser.getFullName() %></h2>

<h2>Email: <jsp:getProperty name="currentUser" property="email" /></h2>

<h2>

Country: <jsp:getProperty name="currentUser" property="country" />

<img src="assets/india.png" width="30px" />

</h2>

<h2>Your UserId is: <%=request.getParameter("userId")%>.</h2>

<p>Please use the id for future references.</p>

<p>

You've will further notifications on your email address at: <%=

currentUser.getMailtoUrl() %>

</p>

</div>

</body>

</html>

// login.jsp

<!-- 1. JSP useBean Action tag -->

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %> <%@page

import="java.util.Random" %>

<jsp:useBean id="currentUser" class="p1.User" scope="application">

</jsp:useBean>

<!-- 2. JSP SetProperty Action Tag -->

<jsp:setProperty name="currentUser" property="\*" />

<!-- Generating a random Integer between 100 - 1000 -->

<% Random r = new Random(); int id = r.nextInt(1000)+100; %>

<!-- 3. JSP forward action tag -->

<c:if test="${currentUser.country == 'india'}">

<jsp:forward page="indian.jsp">

<jsp:param name="userId" value="<%= id %>" /> </jsp:forward

></c:if>

<c:if test="${currentUser.country == 'pakistan'}">

<jsp:forward page="pakistani.jsp">

<jsp:param name="userId" value="<%= id %>" /></jsp:forward

></c:if>

<c:if test="${currentUser.country == 'russia'}">

<jsp:forward page="russian.jsp">

<jsp:param name="userId" value="<%= id %>" /></jsp:forward

></c:if>

// pakistani.jsp

<jsp:useBean id="currentUser" class="p1.User" scope="application">

</jsp:useBean>

<!DOCTYPE html>

<html>

<style>

html {

background-color: #388e3c;

}

.heading {

text-decoration: underline;

}

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 2px;

background-color: #a5d6a7;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

box-shadow: rgba(50, 50, 93, 0.25) 0px 13px 27px -5px,

rgba(0, 0, 0, 0.3) 0px 8px 16px -8px;

}

.bio {

font-weight: 500;

font-size: large;

}

.light {

color: #c8e6c9;

}

</style>

<body>

<h1 class="heading light">User Profile</h1>

<!-- 4. Jsp include action tag -->

<jsp:include page="userbio.jsp"></jsp:include>

<div class="container">

<h2>Name: <%= currentUser.getFullName() %></h2>

<h2>Email: <jsp:getProperty name="currentUser" property="email" /></h2>

<h2>

Country: <jsp:getProperty name="currentUser" property="country" />

<img src="assets/pk.png" width="30px" />

</h2>

<h2>Your UserId is: <%=request.getParameter("userId")%>.</h2>

<p>Please use the id for future references.</p>

<p>

You've will further notifications on your email address at: <%=

currentUser.getMailtoUrl() %>

</p>

</div>

</body>

</html>

// russian.jsp

<jsp:useBean id="currentUser" class="p1.User" scope="application">

</jsp:useBean>

<!DOCTYPE html>

<html>

<style>

html {

background-color: #e3f2fd;

}

.heading {

text-decoration: underline;

}

.heading,

p {

text-align: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.container {

width: 40%;

margin: auto;

}

div {

border-radius: 2px;

background-color: #fbe9e7;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

box-shadow: rgba(50, 50, 93, 0.25) 0px 13px 27px -5px,

rgba(0, 0, 0, 0.3) 0px 8px 16px -8px;

}

.bio {

font-weight: 500;

font-size: large;

}

</style>

<body>

<h1 class="heading">User Profile</h1>

<!-- 4. Jsp include action tag -->

<jsp:include page="userbio.jsp"></jsp:include>

<div class="container">

<h2>Name: <%= currentUser.getFullName() %></h2>

<h2>Email: <jsp:getProperty name="currentUser" property="email" /></h2>

<h2>

Country: <jsp:getProperty name="currentUser" property="country" />

<img src="assets/rus.png" width="30px" />

</h2>

<h2>Your UserId is: <%=request.getParameter("userId")%>.</h2>

<p>Please use the id for future references.</p>

<p>

You've will further notifications on your email address at: <%=

currentUser.getMailtoUrl() %>

</p>

</div>

</body>

</html>

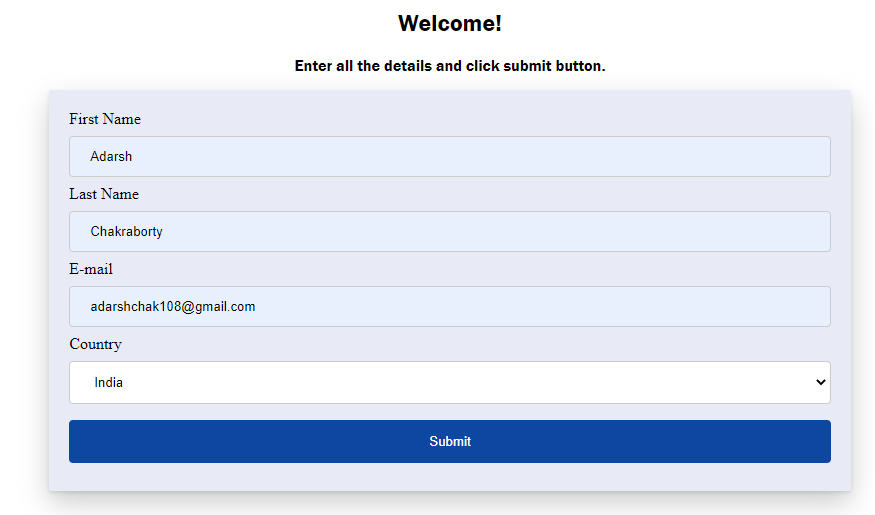
// userbio.jsp

<jsp:useBean id="currentUser" class="p1.User" scope="application">

</jsp:useBean>

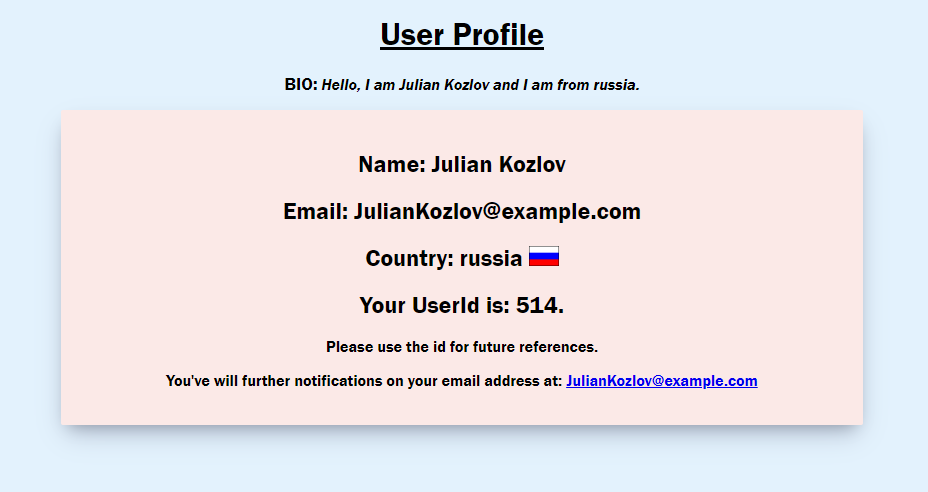
<p><span class="bio light">BIO:</span> <I class=”light”><%= currentUser.getBIO() %></i></p>

**Output - 23**









**Problem 24: Write a JSP program to demonstrate directives.**

// index.jsp

<!-- Page Directives -->

<!-- Include header jsp file -->

<%@include file="header.jsp" %>

<!-- include main content -->

<%@include file="main.jsp" %>

<!-- include footer -->

<%@include file="footer.jsp" %>

// main.jsp

<!-- Importing classes using page directive -->

<%@page import="java.util.Date" %>

<main>

<form action="division.jsp" method="post">

<div class="container">

<h2>Enter number in both fields and click submit</h2>

<div class="main">

<label

>Enter first number:

<input type="text" name="value1" />

</label>

<label>

Enter second number:

<input type="text" name="value2" />

</label>

</div>

<button class="btn" type="submit">submit</button>

</div>

</form>

</main>

// styles.jsp

<style>

html {

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

header {

margin-top: 3rem;

}

main {

margin: 1rem;

}

header {

display: flex;

justify-content: center;

font-weight: 900;

font-size: xx-large;

}

.container {

width: 40%;

margin: auto;

display: flex;

flex-direction: column;

box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;

}

div {

border-radius: 5px;

background-color: #cfd8dc;

padding: 20px;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

text-align: center;

}

footer {

text-align: center;

margin-top: 1rem;

font-family: Arial, Helvetica, sans-serif;

}

.main {

display: flex;

gap: 2rem;

}

.btn {

align-self: center;

max-width: min-content;

padding: 4px 10px;

}

.error {

background-color: #d32f2f;

color: #ffcdd2;

}

.errorContainer {

background-color: #ffcdd2;

color: #c62828;

}

</style>

// division.jsp

<!-- Importing classes using page directive -->

<%@page import="java.util.Date" %>

<!-- Setting an Error Page using page directive -->

<%@page errorPage="error.jsp"%>

<!-- Include styling -->

<%@include file="styles.jsp" %>

<!-- Get both numbers from the request -->

<% int x = Integer.parseInt(request.getParameter("value1")); int y =

Integer.parseInt(request.getParameter("value2")); int result = x/y; %>

<main>

<div class="container">

<h2>Division Result:</h2>

<div class="main" style="justify-content: center">

<%= x %> / <%= y %> = <%= result %>

</div>

</div>

</main>

// error.jsp

<!-- Declrating this as an error page using page directive -->

<%@page isErrorPage="true"%> <%@include file="styles.jsp" %>

<!-- Include styling -->

<%@include file="styles.jsp" %>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Error</title>

</head>

<body>

<main>

<div class="container error">

<h2>An Error occured!</h2>

<div class="main errorContainer" style="justify-content: center">

<h3>The exception is : <%= exception %></h3>

</div>

</div>

</main>

</body>

</html>

// footer.jsp

<footer>

<!-- Printing the developer name using taglib directive -->

<p>This site is made by <strong><c:out value="${developer}"/></strong></p>

<p><strong>Time on the server:</strong> <%=

new Date().toString() %></p>

</footer>

</body>

</html>

// header.jsp

<!-- Including JSTL Custom tag library -->

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!-- Setting an variable using taglib Directive -->

<c:set var="developer" value="Adarsh Chakraborty"></c:set>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Random Number Generator</title>

</head>

<%@include file="styles.jsp" %>

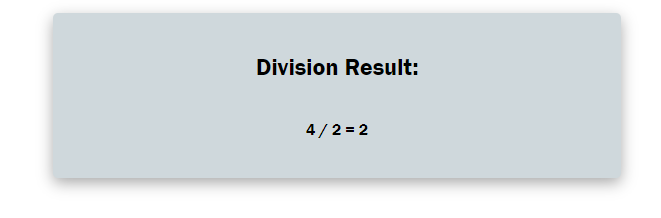
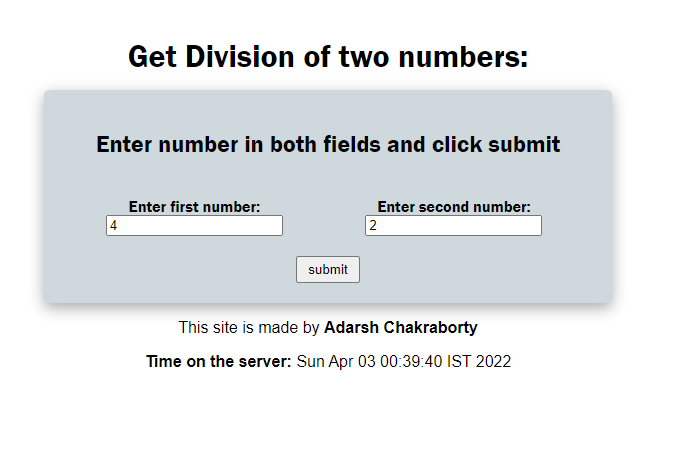
<body>

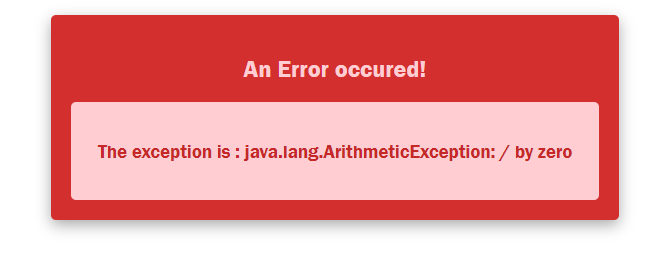
<header>Get Division of two numbers:</header>

</body>

</html>

**Output-24**





**Problem 25: Write a JSP program to demonstrate initial parameter and request parameters access from a JSP page.**

// index.jsp

<% String devName = config.getInitParameter("developer"); %>

<html>

<title>Transport selection</title>

<head>

<style>

body {

display: flex;

justify-content: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.main {

width: 40%;

height: min-content;

border: 1px solid black;

border-radius: 4px;

padding: 2rem 5rem;

margin-top: 1rem;

}

.courses {

display: flex;

gap: .75rem;

}

.transport {

margin-left: 1rem;

display: flex;

flex-direction: column;

gap: 0.5rem;

}

</style>

<body>

<div class="main">

<h2>Select your Courses:</h2>

<form action="output.jsp" method="post">

<div class="courses">

<label>

<input type="radio" checked="checked" name="course" value="BCA">

BCA

</label>

<label>

<input type="radio" name="course" value="MCA">

MCA

</label>

<label>

<input type="radio" name="course" value="MSC">

MSC

</label>

<label>

<input type="radio" name="course" value="BSC">

BSC

</label>

</div>

<h3>Check all your preferred transportation :</h3>

<div class="transport">

<label>

<input type="checkbox" checked="checked" name="transport" value="Bus">

Bus

</label>

<label>

<input type="checkbox" name="transport" value="Train">

Train

</label>

<label>

<input type="checkbox" name="transport" value="Taxi">

Taxi

</label>

<label>

<input type="checkbox" name="transport" value="Private Cab">

Private Cab

</label>

</div>

<br />

<label style="margin-left: 1rem;">

Enter Local address:

<input type="text" name="address" style="width: 40%; margin-left: 1rem;">

</label>

<br />

<br />

<button type="submit" style="margin-left: 1rem; padding: 5px 10px;">Submit</button>

</form>

<hr/>

<div style="text-align: center; padding: 1rem 0;">This site is made by: <%= devName %></div>

<hr/>

</div>

</body>

</html>

// output.jsp

<%!

private String course;

private String[] transport;

private String address;

%>

<%

course = request.getParameter("course");

address = request.getParameter("address");

if(address == null || address == ""){

address = "No Locality was given.";

}

String devName = config.getInitParameter("developer");

transport = request.getParameterValues("transport");

%>

<html>

<title>Transport selection</title>

<head>

<style>

body {

display: flex;

justify-content: center;

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

.main {

width: 40%;

height: min-content;

border: 1px solid black;

border-radius: 4px;

padding: 2rem 5rem;

margin-top: 1rem;

}

.courses {

display: flex;

gap: .75rem;

}

.transport {

margin-left: 1rem;

display: flex;

flex-direction: column;

gap: 0.5rem;

}

</style>

<body>

</body>

<div class="main">

<h2>Request received!!</h2>

<h3>You've selected the following transportation methods for <%= course %> course:</h3>

<%

if (transport!=null)

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

out.println(i+1 + ". " + transport[i] + "<br/>");

if (transport==null)

out.println("No transportation was selected.<br/>");

%>

<p>Locality: <%= address%></p>

<h4>We have received your request, you will be reached out soon.</h4>

<br/>

<hr/>

<div style="text-align: center; padding: 1rem 0;">This site is made by <%= devName %></div>

<hr/>

</div>

</html>

// web.xml

<web-app>

<servlet>

<servlet-name>Index</servlet-name>

<jsp-file>/index.jsp</jsp-file>

<init-param>

<param-name>developer</param-name>

<param-value>Adarsh Chakraborty</param-value>

</init-param>

</servlet>

<servlet>

<servlet-name>Output</servlet-name>

<jsp-file>/output.jsp</jsp-file>

<init-param>

<param-name>developer</param-name>

<param-value>Adarsh Chakraborty</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>Index</servlet-name>

<url-pattern>/index.jsp</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Output</servlet-name>

<url-pattern>/output.jsp</url-pattern>

</servlet-mapping>

</web-app>

**Output – 25**

