**JAVA RECORD**

**WEEK-1:**

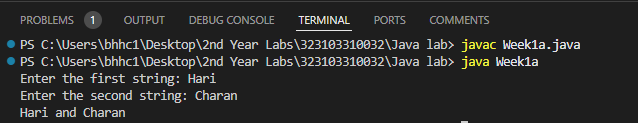
**(i) AIM:** Accept two strings from the user and print them on the console with concatenation of “and” in the middle of the strings.

***PROGRAM:***

**Using the Scanner class method**:

|  |
| --- |
| import java.util.Scanner;  public class Week1a {  public static void main(String args[]) {  Scanner sc = new Scanner(System.in);  String s1, s2;  System.out.print("Enter the first string: ");  s1 = sc.next();  System.out.print("Enter the second string: ");  s2 = sc.next();  System.out.print(s1+" and "+s2);  sc.close();  } } |

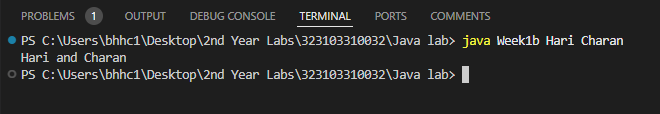
***Output:***

****

**Using Command line arguments:**

|  |
| --- |
| **public class Week1b {  public Week1b() {  }   public static void main(String[] var0) {  String var1 = var0[0];  String var2 = var0[1];  System.out.println(var1 + " and " + var2);  } }** |

***Output:***

****

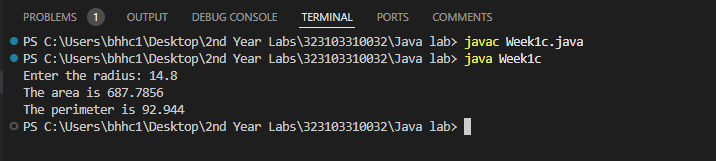
**(ii)AIM**: To find the perimeter and area of a circle given the value of

radius.

***PROGRAM:***

|  |
| --- |
| import java.util.Scanner;  public class Week1c {  public static void main(String args[]) {  Scanner sc = new Scanner(System.in);  double a, p, r;    System.out.print("Enter the radius: ");  r = sc.nextDouble();    a = (3.14) \* (r) \* (r);  p = (2) \* (3.14) \* (r);    System.out.println("The area is " + a);  System.out.println("The perimeter is " + p);  } } |

***Output:***



**WEEK-2:**

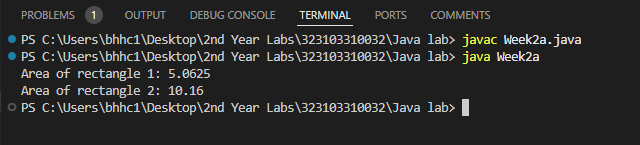
**(a)AIM:** Program to call default constructor first and then any other

constructor in the class.

***PROGRAM:***

|  |
| --- |
| class Rectangle {  double length, breadth;  Rectangle() {  length = 1.25;  breadth = 4.05;  }  Rectangle(double length, double breadth) {  this.length = length;  this.breadth = breadth;  }  double Area() {  return length \* breadth;  } }  class Week2a {  public static void main(String[] args) {  Rectangle r1 = new Rectangle();  Rectangle r2 = new Rectangle(2.54, 4);  System.out.println("Area of rectangle 1: " + r1.Area());  System.out.println("Area of rectangle 2: " + r2.Area());  } } |

***Output:***



**(B) AIM:** Program that accepts an array of integers and print those

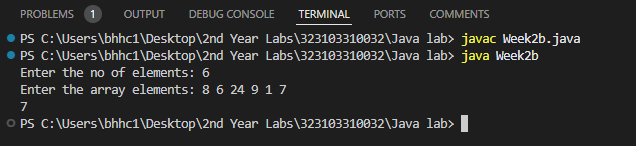
which are both odd and prime. If no such element is in that array print

“Not found”.

***PROGRAM:***

|  |
| --- |
| import java.util.Scanner;  public class Week2b {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);   System.out.print("Enter the no of elements: ");  int n = sc.nextInt();   int[] a = new int[n];  int count = 0, flag = 0;   System.out.print("Enter the array elements: ");  for (int i = 0; i < n; i++) {  a[i] = sc.nextInt();  }   for (int i = 0; i < n; i++) {  for (int j = 1; j <= a[i]; j++) {  if (a[i] % j == 0) {  count++;  }  }  if (count == 2 && a[i] % 2 == 1) {  System.out.print(a[i] + " ");  flag++;  }  count = 0;  }   if (flag == 0) {  System.out.print("Not Found");  }  sc.close();  } } |

***Output:***

******

**(C) AIM:** Program to accept contents into an Integer Array and print

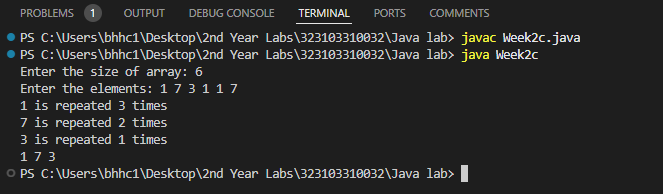
the frequency of each number in the order of their number of

occurrences.

***PROGRAM:***

|  |
| --- |
| import java.util.Scanner;  public class Week2c {  public static void main(String args[]) {  Scanner sc = new Scanner(System.in);    System.out.print("Enter the size of array: ");  int n = sc.nextInt();  int a[] = new int[n];  System.out.println("Enter the elements: ");  for(int i=0; i<n; i++) {  a[i] = sc.nextInt();  }  for(int i=0; i<n; i++) {  int count=1,temp=a[i];  for(int j=i+1; j<n; j++) {  if(a[j]==temp) {  count++;  n--;  for(int k=j; k<n; k++) {  a[k] = a[k+1];  }  j--;  }  }  System.out.println(temp+" is repeated "+count+" times");   }  for(int i=0; i<n; i++) {  System.out.print(a[i]+" ");  }  sc.close();  } } |

***Output:***

******

**(D) AIM:** Program that accepts an “m x n‟ double dimension array,

where “m‟ represents financial years and “n‟ represents Ids of the

items sold. Each element in the array represents number of items

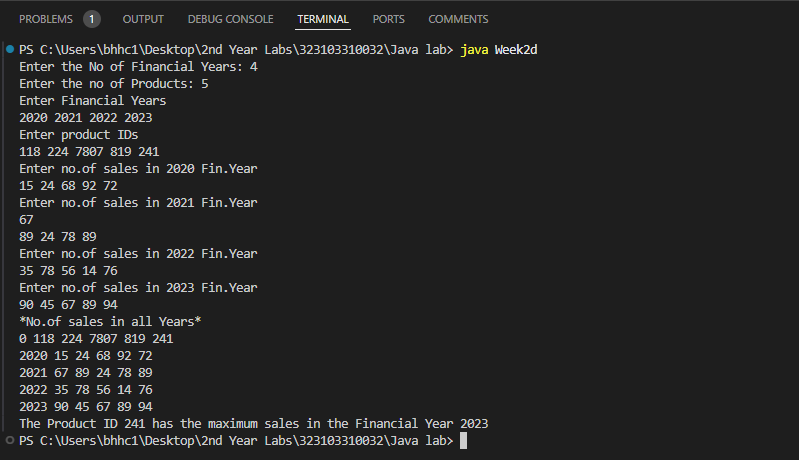
sold in a particular year. Identify the year and id of the item which

has more demand.

***PROGRAM:***

|  |
| --- |
| import java.util.Scanner;  class Week2d {  public static void main(String arg[]) {  int m, n, i, j, fy = 0, pid = 0, max;  Scanner sc = new Scanner(System.in);    System.out.print("Enter the No of Financial Years: ");  m = sc.nextInt();    System.out.print("Enter the no of Products: ");  n = sc.nextInt();    int sales[][] = new int[m + 1][n + 1];    System.out.println("Enter Financial Years");  for (i = 0; i < m; i++) {  sales[i + 1][0] = sc.nextInt();  }    System.out.println("Enter product IDs");  for (j = 0; j < n; j++) {  sales[0][j + 1] = sc.nextInt();  }    for (i = 1; i <= m; i++) {  System.out.println("Enter no.of sales in " + sales[i][0] + " Fin.Year");  for (j = 1; j <= n; j++) {  sales[i][j] = sc.nextInt();  }  }    System.out.print("\*No.of sales in all Years\*\n");  for (i = 0; i <= m; i++) {  for (j = 0; j <= n; j++) {  System.out.print(sales[i][j] + " ");  }  System.out.println();  }    max = sales[1][1];  for (i = 1; i <= m; i++) {  for (j = 1; j <= n; j++) {  if (sales[i][j] > max) {  max = sales[i][j];  fy = sales[i][0];  pid = sales[0][j];  }  }  }    System.out.println("The Product ID " + pid + " has the maximum sales in the Financial Year " + fy);  sc.close();  } } |

***Output:***



**WEEK-3**

**(A) AIM:** Create a class Box that uses a parameterized constructor to

initialize the dimensions of a box. The dimensions of the Box are

width, height, depth. The class should have a method that can return

the volume of the box. Create an object of the Box class and test the

functionalities.

***Program:***

|  |
| --- |
| class Box {  double width, height, depth;   Box(double width, double height, double depth) {  this.width = width;  this.height = height;  this.depth = depth;  }   double getVolume() {  return width \* height \* depth;  } }  class Week3a {  public static void main(String[] args) {  Box b1 = new Box(1.4, 2.25, 1.25);   double x = b1.getVolume();   System.out.println("Volume of the Box is" + " " + x);  } } |

***Output:***



**(B)AIM:** Create a new class called Calculator with the following

methods: A static method called powerInt(int num1,int num2) This

method should return num1 to the power num2. A static method

called powerDouble(double num1,double num2). This method

should return num1 to the power num2. Invoke both the methods

and test the functionality. Also count the number of objects created

***Program:***

|  |
| --- |
| class Week3b {  public static int powerInt(int num1, int num2) {  return (int) Math.pow(num1, num2);  }   public static double powerDouble(double num1, double num2) {  return Math.pow(num1, num2);  }   public static void main(String[] args) {  System.out.println("Output using powerInt method: " + powerInt(25, 3));  System.out.println("Output using powerDouble method: " + powerDouble(2.4, 3));  } } |

***Output:***



**WEEK-4**

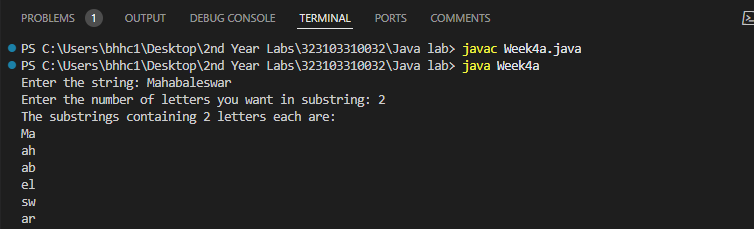
**(A) AIM:** Accept a String and a number “n” from the user. Divide the given

string into substrings each of size “n” and sort them lexicographically

***PROGRAM:***

|  |
| --- |
| import java.util.Scanner; import java.util.Arrays;  public class Week4a {  public static void main(String args[]) {  Scanner sc = new Scanner(System.in);  String s1;  int n;  int count = 0;   System.out.print("Enter the string: ");  s1 = sc.next();   System.out.print("Enter the number of letters you want in substring: ");  n = sc.nextInt();   System.out.println("The substrings containing " + n + " letters each are: ");  for (int i = 0; i < (s1.length() / n) + 1; i++) {  StringBuffer s2 = new StringBuffer();  for (int j = 0; j < n && count < s1.length(); j++) {  s2.append(s1.charAt(count));  count++;  }  String s3 = s2.toString();  char arr[] = s3.toCharArray();  Arrays.sort(arr);  String s4 = new String(arr);  System.out.println(s4);  }  sc.close();  } } |

***Output:***



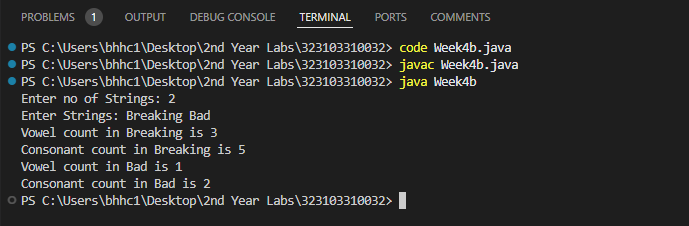
**(B) AIM:** Accept an array of strings and display the number of vowels

and consonants occurred in each string.

***PROGRAM:***

|  |
| --- |
| import java.util.Scanner;  class Week4b {  public static void main(String args[]) {  Scanner sc = new Scanner(System.in);  int n;  System.out.print("Enter no of Strings: ");  n = sc.nextInt();  String str[] = new String[n];  System.out.print("Enter Strings: ");  for (int i = 0; i < n; i++) {  str[i] = sc.next();  }  for (int i = 0; i < n; i++) {  int vow = 0;  int cons = 0;  for (int j = 0; j < str[i].length(); j++) {  char ch = str[i].charAt(j);  if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {  vow++;  } else {  cons++;  }  }  System.out.println("Vowel count in " + str[i] + " is " + vow);  System.out.println("Consonant count in " + str[i] + " is " + cons);  }  } } |

***Output:***



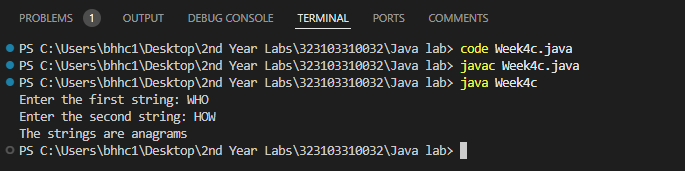
**(C) AIM:** Accept two strings from the user and determine if the strings

are anagrams or not.

***PROGRAM:***

|  |
| --- |
| import java.util.\*;  public class Week4c {  public static void main(String args[]) {  String s1 = new String();  String s2 = new String();  Scanner sc = new Scanner(System.in);    System.out.print("Enter the first string: ");  s1 = sc.next();    System.out.print("Enter the second string: ");  s2 = sc.next();    if (s1.length() == s2.length()) {  char s1ar[] = s1.toCharArray();  char s2ar[] = s2.toCharArray();  Arrays.sort(s1ar);  Arrays.sort(s2ar);    int count = 0;  for (int i = 0; i < s1.length(); i++) {  if (s1ar[i] == s2ar[i]) {  count++;  }  }    if (count == s1.length()) {  System.out.println("The strings are anagrams");  } else {  System.out.println("The strings are not anagrams");  }  } else {  System.out.println("The strings are not anagrams");  }  } } |

***Output:***



**WEEK-5**

**(A) AIM:** Create a multilevel inheritance for classes vehicle, brand and

cost. The vehicle class determines the type of vehicle which is

inherited by the class brand which determines the brand of the

vehicle. Brand class is inherited by cost class, which tells about the

cost of the vehicle. Create another class which calls the constructor

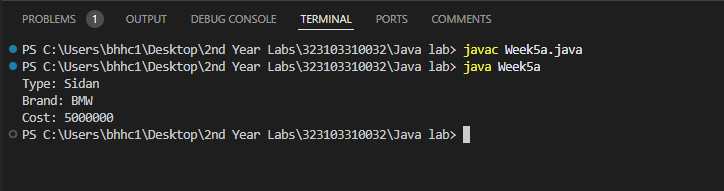
of cost class and method that displays the total vehicle information

from the attributes available in the super classes.

***PROGRAM:***

|  |
| --- |
| class Vehicle{  String type;  Vehicle(String type){  this.type = type;  } }  class Brand extends Vehicle{  String brand;  Brand(String type, String brand){  super(type);  this.brand = brand;  } }  class Cost extends Brand{  int cost;  Cost(String type, String brand, int cost){  super(type, brand);  this.cost = cost;  }  void display(){  System.out.println("Type: "+type);  System.out.println("Brand: "+brand);  System.out.println("Cost: "+cost);  } }    public class Week5a {  public static void main(String[] args) {  Cost c1 = new Cost("Sidan", "BMW", 5000000);  c1.display();  } } |

***Output:***



**(B) AIM:** Create an inheritance hierarchy of Figure\_3D, Cylinder, Cone,

Sphere etc. In the base class provides methods that are common to

all Figure\_3Ds and override these in the derived classes to perform

different behaviors, depending on the specific type of Figure\_3D.

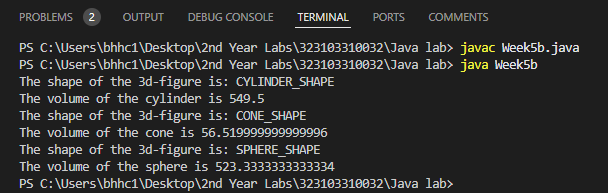
Create an array of Figure\_3D, fill it with different specific types of

Figure\_3Ds and call your base class methods.

***PROGRAM***:

|  |
| --- |
| class Figure\_3d {  String figure;   Figure\_3d(String figure) {  this.figure = figure;  }   void volume() {  return;  } }  class Cylinder extends Figure\_3d {  double height;  double radius;   Cylinder(String figure, double height, double radius) {  super(figure);  this.height = height;  this.radius = radius;  }   @Override  void volume() {  System.out.println("The shape of the 3d-figure is: " + super.figure);  double volume = (3.14) \* radius \* radius \* height;  System.out.println("The volume of the cylinder is " + volume);  } }  class Cone extends Figure\_3d {  double height;  double radius;   Cone(String figure, double height, double radius) {  super(figure);  this.height = height;  this.radius = radius;  }   @Override  void volume() {  System.out.println("The shape of the 3d-figure is: " + figure);  double volume = (3.14) \* radius \* radius \* (height / 3);  System.out.println("The volume of the cone is " + volume);  } }  class Sphere extends Figure\_3d {  double radius;   Sphere(String figure, double radius) {  super(figure);  this.radius = radius;  }   @Override  void volume() {  System.out.println("The shape of the 3d-figure is: " + figure);  double volume = (4 / 3.0) \* (3.14) \* radius \* radius \* radius;  System.out.println("The volume of the sphere is " + volume);  } }  public class Week5b {  public static void main(String[] args) {  Figure\_3d arr[] = new Figure\_3d[3];  Cylinder c1 = new Cylinder("CYLINDER\_SHAPE", 7, 5);  arr[0] = c1;  Cone cn1 = new Cone("CONE\_SHAPE", 6, 3);  arr[1] = cn1;  Sphere s1 = new Sphere("SPHERE\_SHAPE", 5);  arr[2] = s1;  for (int i = 0; i < 3; i++) {  arr[i].volume();  }  } } |

***Output:***



**WEEK-6**

1. **AIM:** Design a package to contain the class Student that contains data members such as name, roll number and another package contains the interface Sports which contains some sports information. Import these two packages in a package called Report which process both Student and Sport and give the report.

**PROGRAM:**

Contents of ***student/Student.java:***

**package** student;

**public** **class** **Student** {

**public** String name;

**public** **int** rollNumber;

**public** **Student**(String name,**int** rollNumber){

**this**.name = name;

**this**.rollNumber = rollNumber;

}

**public** String **getName**(){

**return** name;

}

**public** **int** **getRollNumber**(){

**return** rollNumber;

}

}

Contents of ***sports/Cricket.java:***

**package** sports;

**public** **class** **Cricket** **implements** spo {

**@Override**

**public** String **getSportName**(){

**return** "Cricket";

}

}

Contents of ***sport/Football.java:***

**package** sports;

**public** **class** **Football** **implements** spo {

**@Override**

**public** String **getSportName**(){

**return** "Football";

}

}

Contents of ***sports/Hockey.java:***

**package** sports;

**public** **class** **Hockey** **implements** spo {

**@Override**

**public** String **getSportName**(){

**return** "Hockey";

}

}

Contents of ***sport/Spo.java:***

**package** sports;

**public** **interface** **spo** {

String **getSportName**();

}

Contents of ***Week6a.java:***

**import** **report.rep**;

**import** **sports.Cricket**;

**import** **sports.Football**;

**import** **sports.Hockey**;

**import** **student.Student**;

**public** **class** **Week6a** {

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

// 1st report for student s1 who plays cricket

Student s1 = **new** Student("Charan", **32**);

Cricket cri = **new** Cricket();

rep r1 = **new** rep();

r1.generateReport(s1, cri);

// 2nd report for student s2 who plays football

Student s2 = **new** Student("Vivek", **47**);

Football f = **new** Football();

rep r2 = **new** rep();

r2.generateReport(s2, f);

Student s3 = **new** Student("Bob" , **68**);

Hockey h = **new** Hockey();

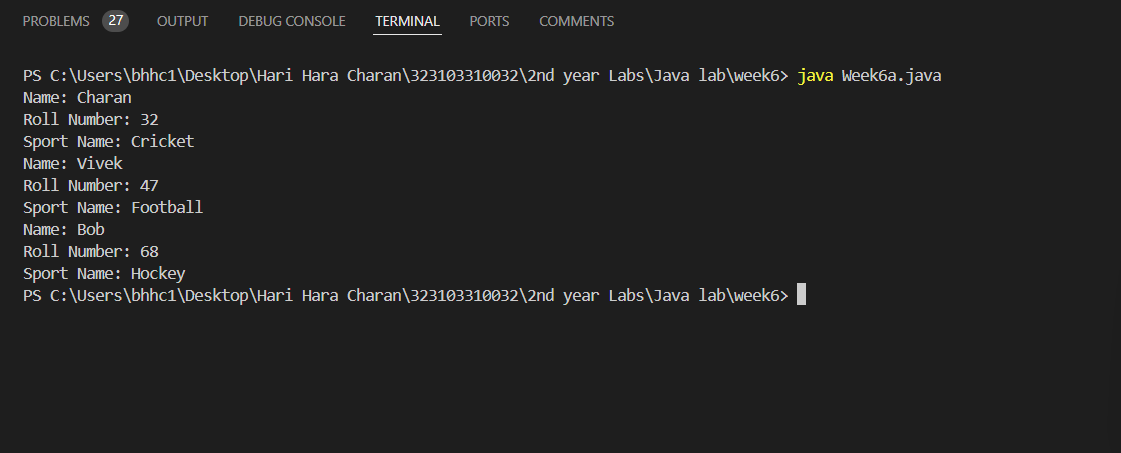
rep r3 = **new** rep();

r3.generateReport(s3, h);

}

}

**OUTPUT:**



1. **AIM:** Write a program that accepts values of different data types and convert them to corresponding wrapper classes and display using the vector

***PROGRAM***:

**import** **java.util.Scanner**;

**import** **java.util.Vector**;

**public** **class** **Week6b** {

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

Scanner sc = **new** Scanner(System.in);

System.out.print("Enter an integer: ");

Integer i = sc.nextInt();

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

Byte by = sc.nextByte();

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

Short sh = sc.nextShort();

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

Long l = sc.nextLong();

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

Float f = sc.nextFloat();

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

Double d = sc.nextDouble();

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

Character c = sc.next().charAt(**0**);

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

String s = sc.next();

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

Boolean b = sc.nextBoolean();

Vector<Object> v = **new** Vector<>();

v.add(i);

v.add(by);

v.add(sh);

v.add(l);

v.add(f);

v.add(d);

v.add(c);

v.add(s);

v.add(b);

**for**(Object item : v) {

System.out.println(item);

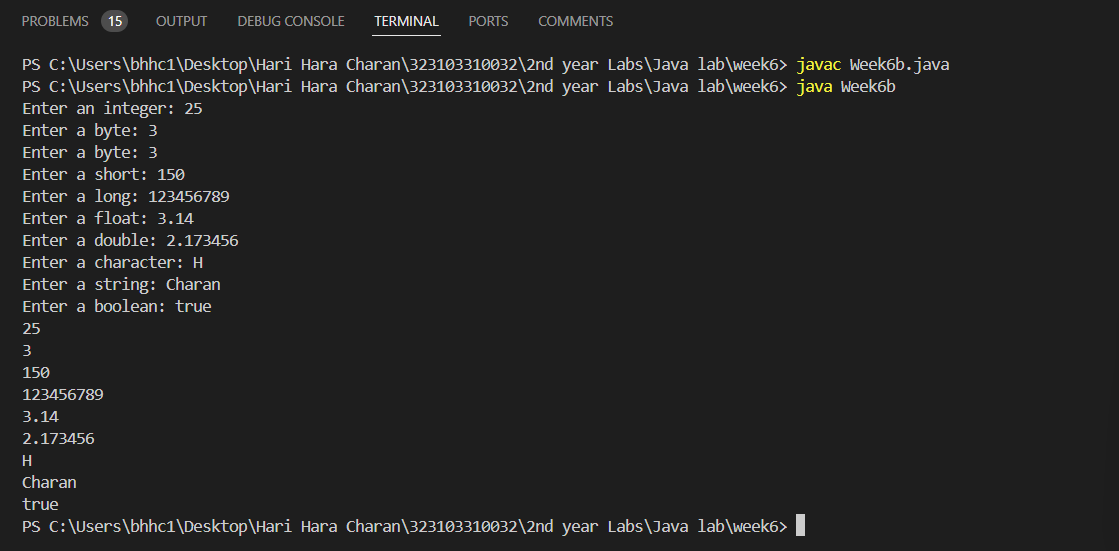
}

sc.close();

}

}

**OUTPUT:**



**WEEK-7**

1. **AIM:** Write a program to generate a set of random numbers between two numbers x1 and x2, and x1>0.

**PROGRAM:**

**import** **java.util.Random**;

**import** **java.util.Scanner**;

**public** **class** **Week7a** {

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

Scanner sc = **new** Scanner(System.in);

Random rand = **new** Random();

System.out.println("enter the value of x1 (x1>0): ");

**int** x1 = sc.nextInt();

**if**(x1 < **0**){

System.out.println("x1 must be grater than 0");

}

System.out.println("enter the value of x2 : ");

**int** x2 = sc.nextInt();

**if**(x2 < x1){

System.out.println("x2 must be grater than x1");

sc.close();

**return**;

}

System.out.println("how many random numbers do you want to generate: ");

**int** count = sc.nextInt();

System.out.print("Generate random numbers between "+ x1 + " and " + x2 + ": ");

**for**(**int** i = **0**; i < count; i++) {

**int** x = rand.nextInt(x2-x1+**1**) + x1;

System.out.print(" " + x);

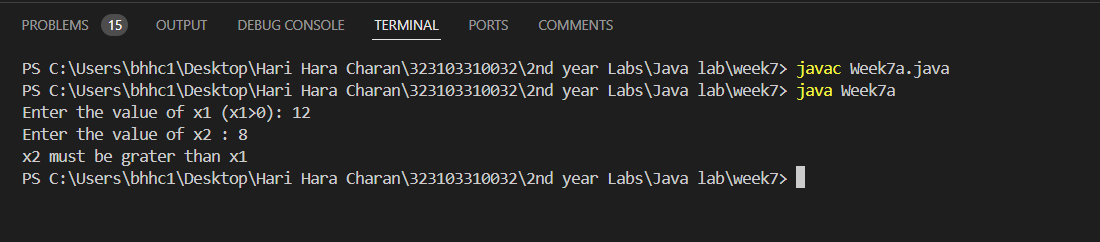
}

sc.close();

}

}

**OUTPUT:**

****

**(B)AIM:** Write a program to implement a new ArrayList class. It should contain add(), get(), remove(), size() methods. Use dynamic array logic.

**PROGRAM:**

**public** **class** **Array** {

**int**[] array;

**int** size;

**public** **Array**() {

array = **new** **int**[**10**];

size = **0**;

}

**public** **void** **add**(**int** x){

**if**(size == array.length){

resize();

}

array[size] = x;

size++;

}

**public** **int** **get**(**int** index){

**if**(index < **0** || index >= size){

**throw** **new** **ArrayIndexOutOfBoundsException**("index out of bounds");

}

**return** array[index];

}

**public** **void** **remove**(**int** index){

**if**(index < **0** || index >= size){

**throw** **new** **ArrayIndexOutOfBoundsException**("index out of bounds");

}

**for**(**int** i = index; i < size-**1**; i++){

array[i] = array[i+**1**];

}

size--;

}

**public** **int** **size**(){

**return** size;

}

**public** **void** **resize**(){

**int**[] newArray = **new** **int**[array.length\***2**];

**for**(**int** i = **0**; i < size; i++){

newArray[i] = array[i];

}

array = newArray;

}

**public** **void** **display**(){

**for**(**int** i = **0**; i < size; i++){

System.out.print(array[i] + " ");

}

}

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

Array list = **new** Array();

list.add(**1**);

list.add(**2**);

list.add(**3**);

list.add(**4**);

list.add(**5**);

System.out.println(list.get(**1**));

list.remove(**2**);

System.out.println(list.size());

list.display();

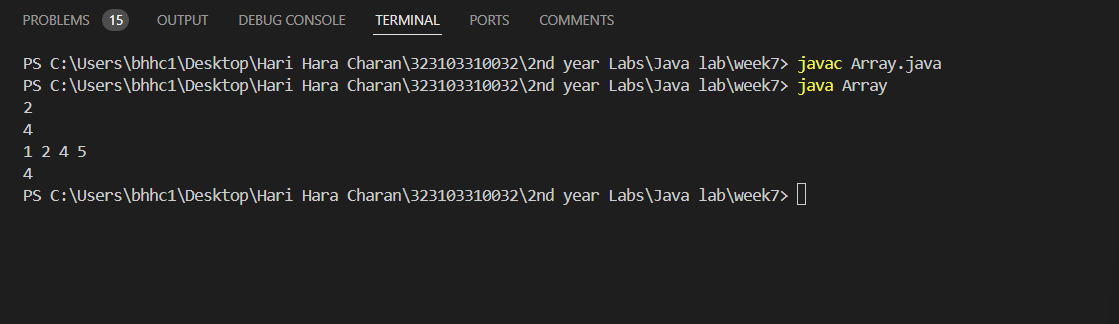
System.out.println();

System.out.println(list.get(**2**));

}

}

Output:

****

**(C)AIM:** Create an employee class containing at least 3 details along with Id, setters, and getters. Insert the employee objects dynamically key as employee id and value as it’s corresponding object into a HashMap. Perform Id based search operation on the HashMap.

**PROGRAM:**

**import** **java.util.HashMap**;

**import** **java.util.Scanner**;

**class** **Employee** {

**private** **int** id;

**private** String name;

**private** String department;

**private** **int** salary;

**public** **Employee**(**int** id, String name, String department, **int** salary) {

**this**.id = id;

**this**.name = name;

**this**.department = department;

**this**.salary = salary;

}

**public** **int** **getId**(){

**return** id;

}

**public** **void** **setId**(**int** id) {

**if**(id>**0** || id<**10**){

**this**.id = id;

}

**else** {

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

}

}

**public** String **getName**(){

**return** name;

}

**public** **void** **setName**(String name){

String[] names = name.split("[0123456789]");

**if**(names.length == **1**){

**this**.name = name;

}

**else**{

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

}

}

**public** String **getDepartment**(){

**return** department;

}

**public** **void** **setDepartment**(String department){

String[] names = name.split("[0123456789]");

**if**(names.length == **1**){

**this**.department = department;

}

**else**{

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

}

}

**public** **int** **getSalary**(){

**return** salary;

}

**public** **void** **setSalary**(**int** salary){

**this**.salary = salary;

}

**public** **void** **display**(){

System.out.println("Employee ID: "+id);

System.out.println("Employee Name: "+name);

System.out.println("Employee Department: "+department);

System.out.println("Employee Salary: "+salary);

}

}

**public** **class** **Week7c** {

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

HashMap<Integer, Employee> employees = **new** HashMap<>();

Scanner sc = **new** Scanner(System.in);

System.out.println("Enter the number of employees: ");

**int** n = sc.nextInt();

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

System.out.println("enter details for employee #" + (i + **1**));

System.out.println("Id: ");

**int** id = sc.nextInt();

System.out.println("Name: ");

String name = sc.next();

System.out.println("Department: ");

String department = sc.next();

System.out.println("Salary: ");

**int** salary = sc.nextInt();

Employee employee = **new** Employee(id, name, department, salary);

employees.put(id, employee);

}

System.out.println("Enter the employee ID: ");

**int** searchId = sc.nextInt();

**if**(employees.containsKey(searchId)){

System.out.println("Employee found");

Employee found = employees.get(searchId);

found.display();

}

**else**{

System.out.println("Employee not found");

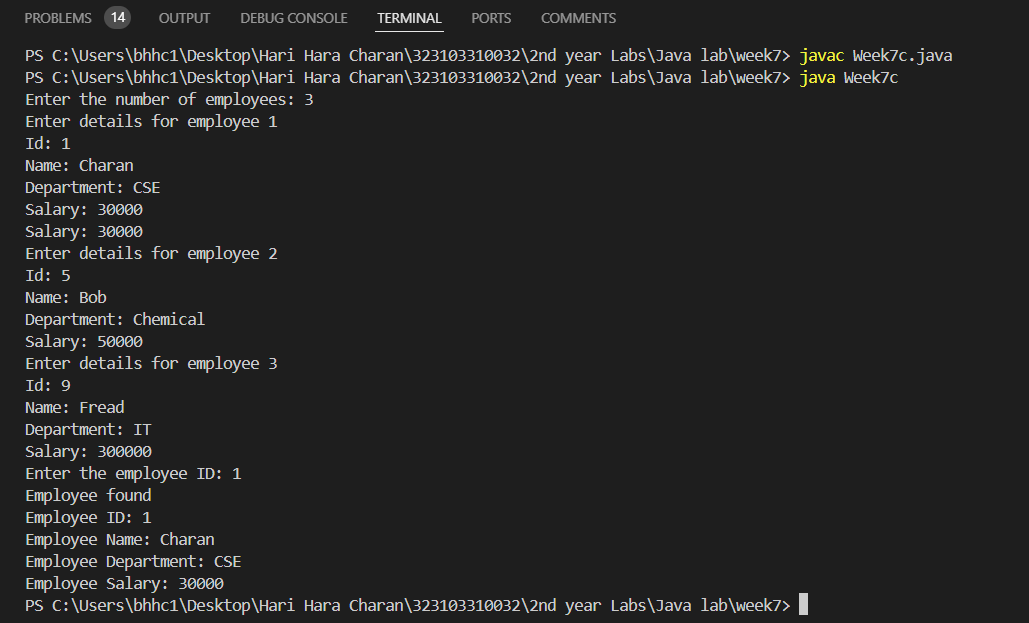
}

sc.close();

}

}

**OUTPUT:**

****

**WEEK-8**

**(A)AIM:** Write a program that reads file name from the user then displays information about that file, also read the contents from the file in byte stream to count the number of alphabets, numeric values, and special symbols. Write these statistics into another file using byte streams

**PROGRAM:**

**import** **java.util.Scanner**;

**import** **java.io.File**;

**import** **java.io.FileInputStream**;

**import** **java.io.FileOutputStream**;

**import** **java.io.IOException**;

**public** **class** **FileAnalyzer** {

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

Scanner sc = **new** Scanner(System.in);

System.out.print("Enter path of the file to be analyzed: ");

String filename = sc.nextLine();

File file = **new** File(filename);

**if** (file.exists()) {

System.out.println("File name "+file.getName());

System.out.println("absolute path "+file.getAbsolutePath());

System.out.println("writeable "+file.canWrite());

System.out.println("readable "+file.canRead());

System.out.println("file size in bytes "+file.length());

}

**else** {

System.out.println("File not found");

}

**int** alphabets = **0**;

**int** digits = **0**;

**int** special = **0**;

**try**(FileInputStream fis = **new** FileInputStream(filename)){

**int** bytedata;

**while**((bytedata = fis.read())!=-**1**){

**if**((bytedata >= 'A' && bytedata <= 'Z') || (bytedata >= 'a' && bytedata <= 'z')){

alphabets++;

}

**else** **if**((bytedata >= '0' && bytedata <= '9')){

digits++;

}

**else**{

special++;

}

}

}

**catch**(IOException e){

System.out.println(e.getMessage());

}

System.out.println("Alphabets: " + alphabets);

System.out.println("Digits: " + digits);

System.out.println("Special: " + special);

String outputfilename = "example1.txt";

**try** (FileOutputStream fos = **new** FileOutputStream(outputfilename)) {

String result = "Alphabets: " + alphabets + "\n" +

"Digits: " + digits + "\n" +

"Special Characters: " + special + "\n";

fos.write(result.getBytes());

System.out.println("Statistics written to " + outputfilename);

} **catch** (IOException e) {

System.out.println("Error writing to the file: " + e.getMessage());

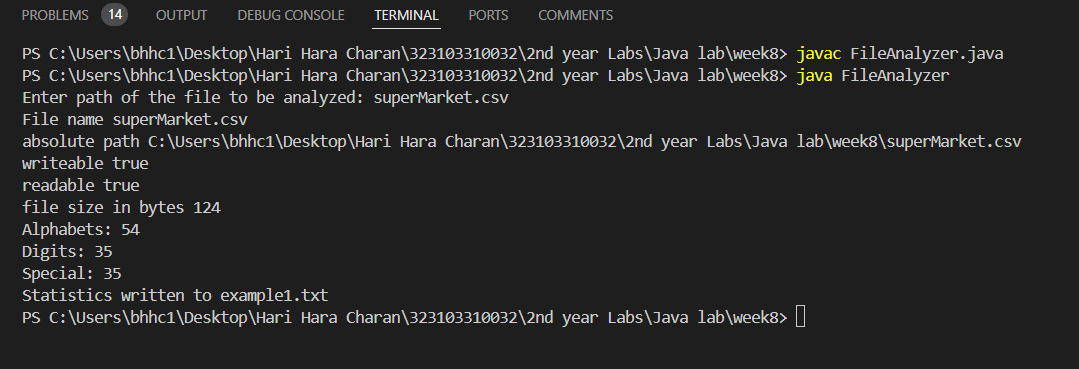
}

sc.close();

}

}

**OUTPUT:**

****

**(B)AIM:** Write a program that reads a CSV file containing a super market data containing product ID, Name, Cost and Quantity of sales and calculate the total revenue of the supermarket also sort the products in the order of their demand.

**PROGRAM:**

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.Scanner;

class Week8b{

public static void main(String arg[]) throws FileNotFoundException, IOException

{

Scanner s=new Scanner(new File("market.csv"));

int total\_revenue=0,i=0;

String[][] products=new String[10][4];

while(s.hasNext())

{

products[i]=s.next().split(",");

total\_revenue+=Integer.parseInt(products[i][2])\*Integer.parseInt(products[i][3]);

i+=1;

}

System.out.println("Total revenue of the Super Market is: "+total\_revenue);

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

for(int y=x+1;y<i;y++)

{

if(Integer.parseInt(products[y][3])>Integer.parseInt(products[x][3]))

{

String temp[]=products[y];

products[y]=products[x]; products[x]=temp;

}

}

}

FileOutputStream out=new FileOutputStream("market.csv");

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

for(int y=0;y<4;y++)

{

out.write((products[x][y]+",").getBytes());

}

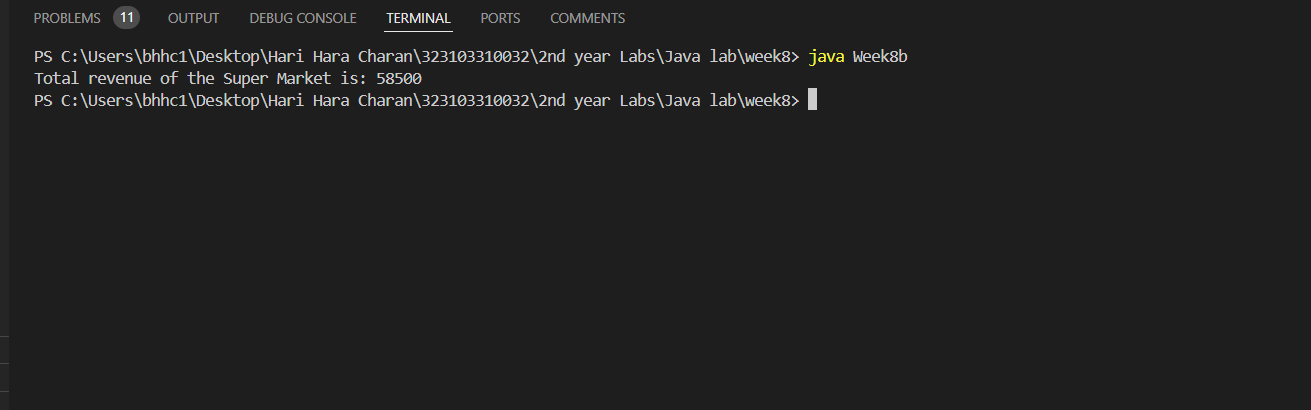
out.write("\n".getBytes());

}

}

}

**OUTPUT:**



**(C)AIM:** Write a program that reads a text file containing some technical content and identify the technical terms and sort them alphabetically. Note: use a file containing stop words (general English and Grammar terms as many as possible)

**PROGRAM:**

import java.io.BufferedReader;

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.Collections;

import java.util.List;

import java.util.Scanner;

class Week8c{

public static

void main(String arg[])

throws IOException,FileNotFoundException

{

Scanner s=new Scanner(System.in);

System.out.print("Enter the file name containing technical content: ");

File techFile=new File(s.next());

System.out.print("Enter the file name containing stop words: ");

File stopWordsFile=new File(s.next());

BufferedReader f1=new BufferedReader(new FileReader(techFile));

BufferedReader f2=new BufferedReader(new FileReader(stopWordsFile));

List<String> stopWords = new ArrayList<>();

List<String> techWords = new ArrayList<>();

String line;

while((line=f2.readLine()) != null)

{

for(String x:line.split(" "))

{

stopWords.add(x);

}

}

while((line=f1.readLine()) !=null)

{

for(String x:line.split(" "))

{

if(!stopWords.contains(x))

techWords.add(x);

}

}

f1.close();

f2.close();

Collections.sort(techWords);

FileWriter fw=new FileWriter(techFile);

for(String word:techWords)

{

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

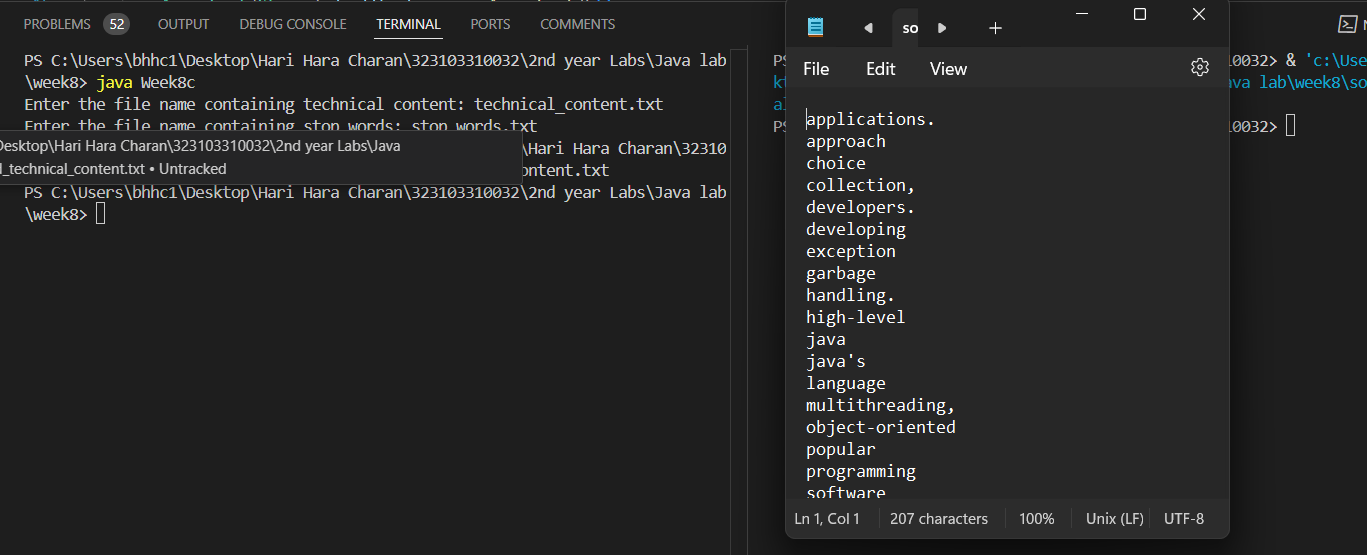
}

fw.close();

}

}

**OUTPUT:**

****

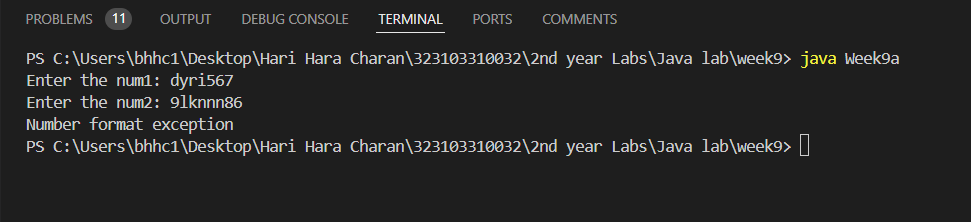
**WEEK-9**

**(A) AIM:** To write a program that reads two numbers from the user to perform integer division into Num1 and Num2 variables. The division of Num1 and Num2 is displayed if they are integers. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception

***Program:***

|  |
| --- |
| import java.util.Scanner;  public class Week9a {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.print("Enter the num1: ");  String num1 = sc.nextLine();  System.out.print("Enter the num2: ");  String num2 = sc.nextLine();  if (num2.equals("0")) {  System.out.println("Divide by zero");  } else {  try {  int input1 = Integer.parseInt(num1);  int input2 = Integer.parseInt(num2);  int result = input1 / input2;  System.out.println("result of " + input1 + "/" + input2 + " = " + result);  } catch (NumberFormatException e) {  System.out.println("Number format exception");  } catch (ArithmeticException e) {  System.out.println("Arithmetic exception");  } finally {  sc.close();  }  }  }  } |

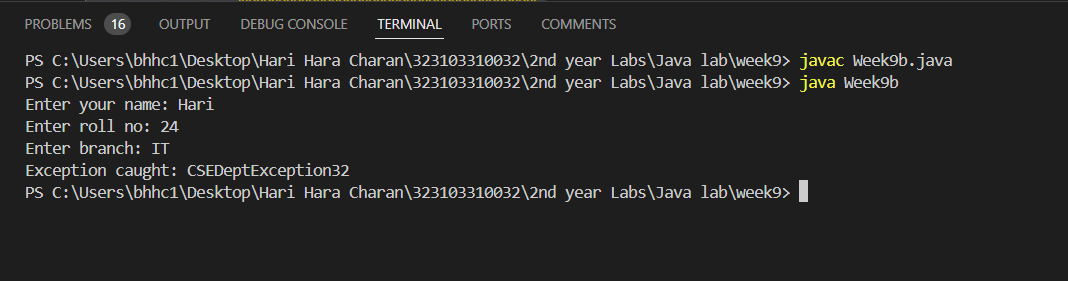
***Output:***

****

**(B) AIM:** To create a user defined exception

|  |
| --- |
| ***Program:***  import java.util.Scanner;  class CSEDeptException32 extends Exception {}  class Week9b {  public static void main(String arg[]) {  Scanner s = new Scanner(System.in);  System.out.print("Enter your name: ");  String name = s.nextLine();  System.out.print("Enter roll no: ");  int rno = s.nextInt();  System.out.print("Enter branch: ");  String branch = s.next();    try {  if (branch.equals("CSE")) {  System.out.println("Name: " + name + " \nRoll no: " + rno);  } else {  throw new CSEDeptException32();  }  } catch (CSEDeptException32 e) {  System.out.println("Exception caught: " + e);  }  s.close();  }  } |

***Output:***

****

**WEEK-10**

**(A) AIM:** To write a program that creates 3 threads by extending the Thread class. First thread displays “Good Morning” every 1 sec, the second thread displays “Hello” every 2 seconds and the third displays Welcome” every 3 seconds. (Repeat the same by implementing Runnable).

***Program:***

class Thread1 extends Thread {

public void run() {

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(1000);

System.out.println("Good Morning");

}

System.out.println("Thread 1 Exited");

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

class Thread2 extends Thread {

public void run() {

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(2000);

System.out.println("Hello");

}

System.out.println("Thread 2 Exited");

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

class Thread3 extends Thread {

public void run() {

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(3000);

System.out.println("Welcome");

}

System.out.println("Thread 3 Exited");

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

public class Week10a {

public static void main(String[] args) {

new Thread1().start();

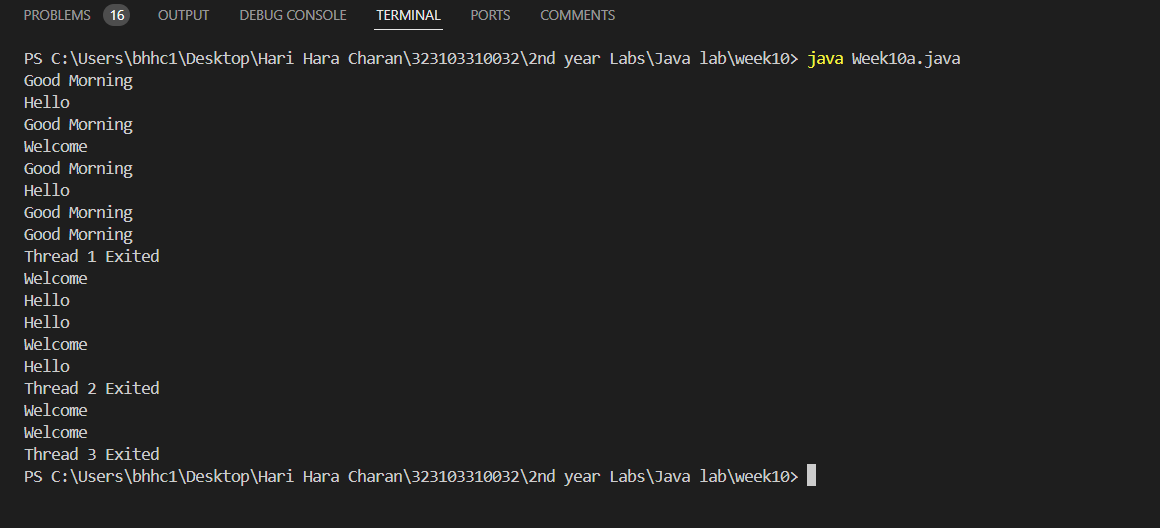
new Thread2().start();

new Thread3().start();

}

}

***Output:***



**(A2) AIM:** To write a program that creates 3 threads by extending the Thread class. First thread displays “Good Morning” every 1 sec, the second thread displays “Hello” every 2 seconds and the third displays Welcome” every 3 seconds using Runnable.

.

***Program:***

class Runnable1 implements Runnable {

Thread t;

Runnable1() {

t = new Thread(this);

}

public void run() {

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(1000);

System.out.println("Good Morning");

}

System.out.println("Thread 1 Exited");

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

class Runnable2 implements Runnable {

Thread t;

Runnable2() {

t = new Thread(this);

}

public void run() {

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(2000);

System.out.println("Hello");

}

System.out.println("Thread 2 Exited");

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

class Runnable3 implements Runnable {

Thread t;

Runnable3() {

t = new Thread(this);

}

public void run() {

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(3000);

System.out.println("Welcome");

}

System.out.println("Thread 3 Exited");

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

class Week10b {

public static void main(String arg[]) {

new Runnable1().t.start();

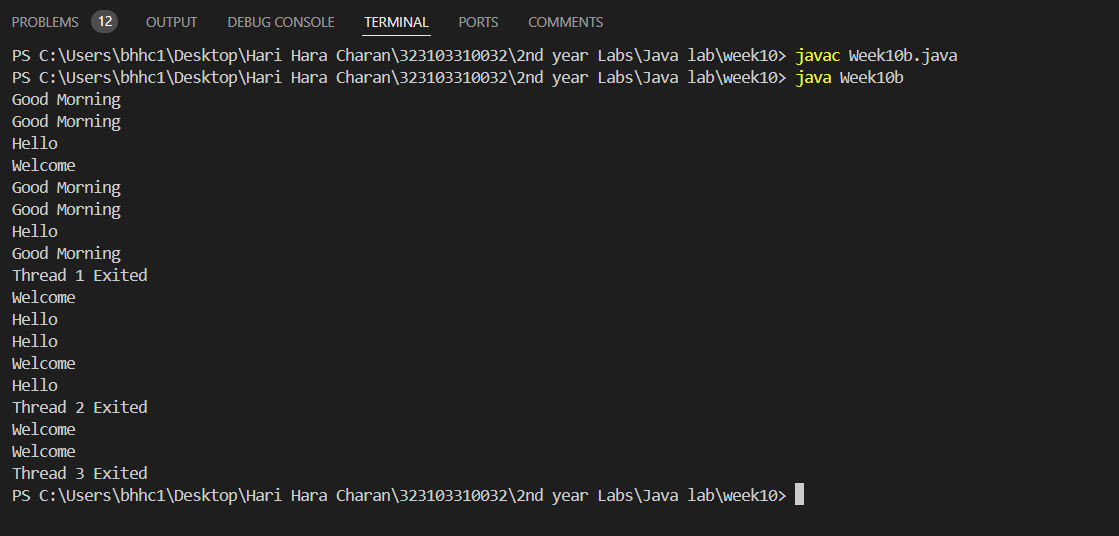
new Runnable2().t.start();

new Runnable3().t.start();

}

}

***Output:***



**(B) AIM:** To write a program to illustrate Thread synchronization

***Program:***

class xyz {

synchronized static void printnum(int n) {

try {

for (int i = 1; i <= 5; i++) {

System.out.println(n \* i);

Thread.sleep(1000);

}

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

class t1 extends Thread {

public void run() {

xyz.printnum(10);

}

}

class t2 extends Thread {

public void run() {

xyz.printnum(1);

}

}

public class Week10b {

public static void main(String[] args) {

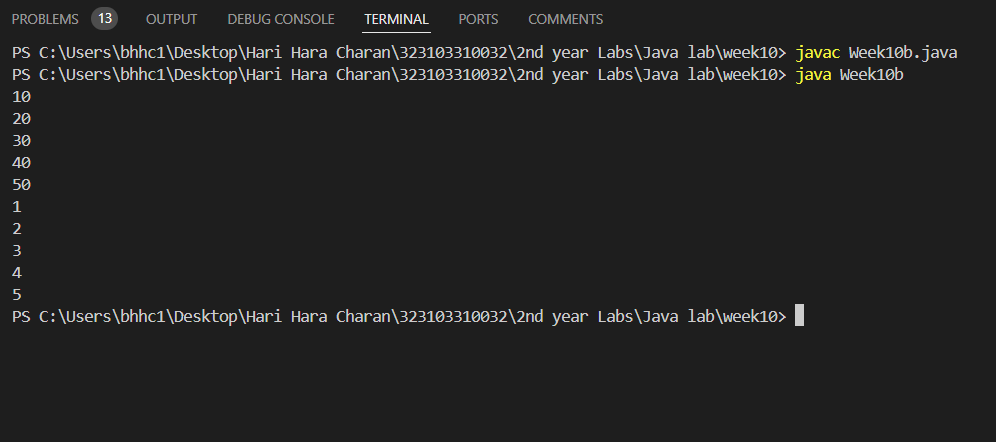
new t1().start();

new t2().start();

}

}

***Output:***



**WEEK-11**

1. **AIM:** To create a JApplet that displays a message which is scrolling from left to right

***Program:***

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.Timer;

import java.awt.Graphics;

import java.awt.Font;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Week11a extends JPanel {

private int xPosition = 0;

private String message = "Hello, Welcome to My Application!";

private Timer timer;

public Week11a() {

setBackground(Color.WHITE);

setFont(new Font("Arial", Font.BOLD, 24));

timer = new Timer(50, new ActionListener() {

public void actionPerformed(ActionEvent e) {

xPosition += 5;

if (xPosition > getWidth()) {

xPosition = -getFontMetrics(getFont()).stringWidth(message);

}

repaint();

}

});

timer.start();

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g); // Clear the background

g.setColor(Color.BLUE); // Set text color

g.drawString(message, xPosition, getHeight() / 2);

}

public static void main(String[] args) {

JFrame frame = new JFrame("Scrolling Message");

Week11a panel = new Week11a();

frame.add(panel);

frame.setSize(600, 200);

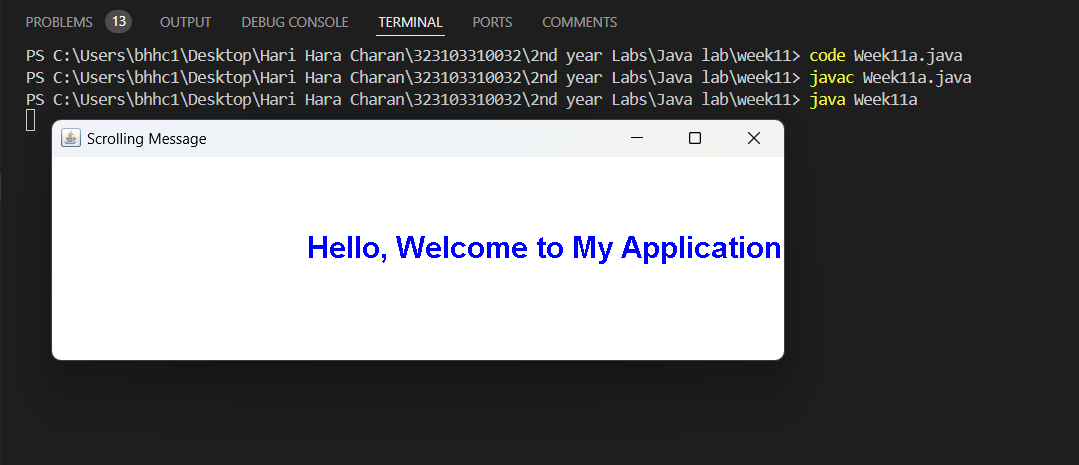
frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

}

***Output:***



**(B)AIM:** To Write a program that displays a sample registration page using Swing controls use appropriate layout managers.

***Program:***

import java.awt.\*;

import javax.swing.\*;

public class Week11b {

public static void main(String[] args) {

JFrame frame = new JFrame("Registration Page");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(400, 300);

frame.setLocationRelativeTo(null);

JPanel panel = new JPanel(new GridLayout(6, 2, 10, 10)); // 6 rows, 2 columns, spacing 10px

panel.add(new JLabel("First Name:"));

JTextField firstNameField = new JTextField();

panel.add(firstNameField);

panel.add(new JLabel("Last Name:"));

JTextField lastNameField = new JTextField();

panel.add(lastNameField);

panel.add(new JLabel("Email:"));

JTextField emailField = new JTextField();

panel.add(emailField);

panel.add(new JLabel("Password:"));

JPasswordField passwordField = new JPasswordField();

panel.add(passwordField);

panel.add(new JLabel("Confirm Password:"));

JPasswordField confirmPasswordField = new JPasswordField();

panel.add(confirmPasswordField);

JButton registerButton = new JButton("Register");

JButton clearButton = new JButton("Clear");

JPanel buttonPanel = new JPanel();

buttonPanel.add(registerButton);

buttonPanel.add(clearButton);

frame.add(panel, BorderLayout.CENTER);

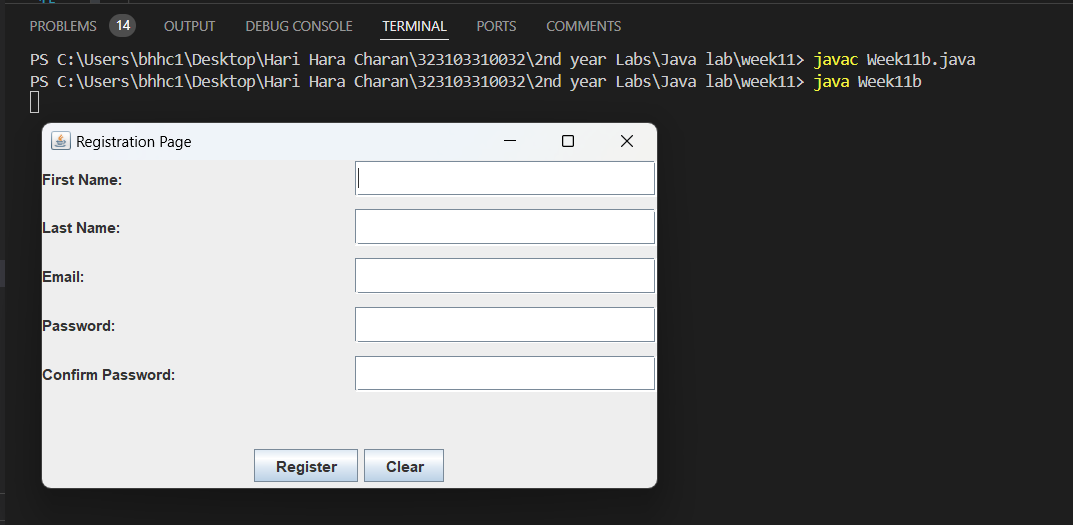
frame.add(buttonPanel, BorderLayout.SOUTH);

frame.setVisible(true);

}

}

***Output:***



**(C) AIM:** To Write a program for handling mouse events with adapter classes.

***Program:***

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class MouseEventAdapterExample extends JFrame {

private JLabel statusLabel;

public MouseEventAdapterExample() {

setTitle("Mouse Event Example");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

statusLabel = new JLabel("Interact with the panel to see mouse events.");

statusLabel.setHorizontalAlignment(SwingConstants.CENTER);

add(statusLabel, BorderLayout.SOUTH);

JPanel panel = new JPanel();

panel.setBackground(Color.LIGHT\_GRAY);

add(panel, BorderLayout.CENTER);

panel.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

statusLabel.setText("Mouse clicked at (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseEntered(MouseEvent e) {

statusLabel.setText("Mouse entered the panel.");

}

public void mouseExited(MouseEvent e) {

statusLabel.setText("Mouse exited the panel.");

}

public void mouseReleased(MouseEvent e) {

statusLabel.setText("Mouse released at (" + e.getX() + ", " + e.getY() + ")");

}

});

panel.addMouseMotionListener(new MouseAdapter() {

@Override

public void mouseMoved(MouseEvent e) {

statusLabel.setText("Mouse moved to (" + e.getX() + ", " + e.getY() + ")");

}

});

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

MouseEventAdapterExample frame = new MouseEventAdapterExample();

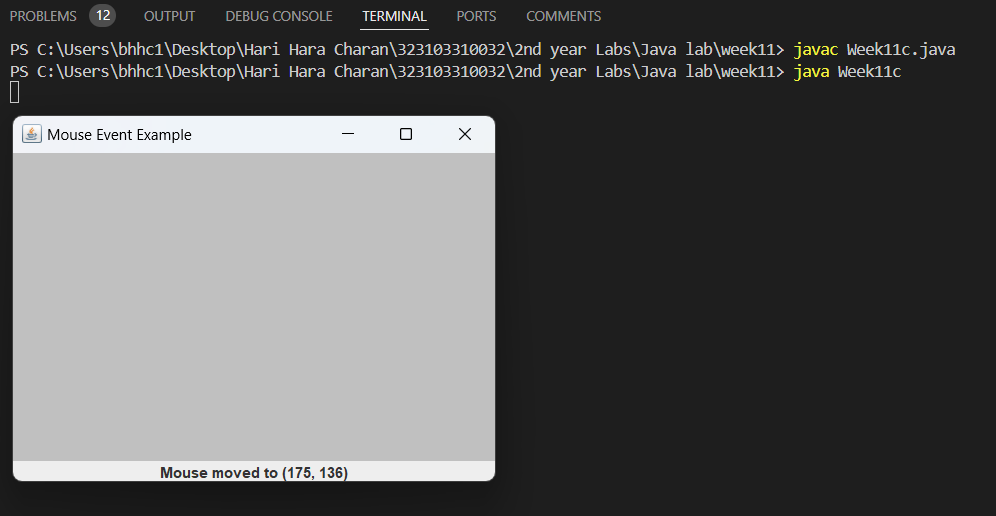
frame.setVisible(true);

});

}

}

***Output:***



**WEEK-12**

**(A) AIM:** To create an interface containing 3 radio buttons named line, rectangle, and oval. Based on the radio button selected, allow user to draw lines, rectangles, or ovals as per the locations selected by the user.

***Program:***

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Week12a extends JFrame {

private JRadioButton lineButton, rectangleButton, ovalButton;

private JPanel drawingPanel;

private String selectedShape = "Line";

private Point startPoint, endPoint;

public Week12a() {

setTitle("Shape Drawer");

setSize(600, 400);

setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

setLayout(new BorderLayout());

lineButton = new JRadioButton("Line");

rectangleButton = new JRadioButton("Rectangle");

ovalButton = new JRadioButton("Oval");

ButtonGroup group = new ButtonGroup();

group.add(lineButton);

group.add(rectangleButton);

group.add(ovalButton);

lineButton.setSelected(true);

JPanel controlPanel = new JPanel();

controlPanel.setLayout(new FlowLayout());

controlPanel.add(lineButton);

controlPanel.add(rectangleButton);

controlPanel.add(ovalButton);

add(controlPanel, BorderLayout.NORTH);

drawingPanel = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

if (startPoint != null && endPoint != null) {

Graphics2D g2d = (Graphics2D) g;

g2d.setStroke(new BasicStroke(2));

switch (selectedShape) {

case "Line":

g2d.drawLine(startPoint.x, startPoint.y, endPoint.x, endPoint.y);

break;

case "Rectangle":

int rectX = Math.min(startPoint.x, endPoint.x);

int rectY = Math.min(startPoint.y, endPoint.y);

int rectWidth = Math.abs(endPoint.x - startPoint.x);

int rectHeight = Math.abs(endPoint.y - startPoint.y);

g2d.drawRect(rectX, rectY, rectWidth, rectHeight);

break;

case "Oval":

int ovalX = Math.min(startPoint.x, endPoint.x);

int ovalY = Math.min(startPoint.y, endPoint.y);

int ovalWidth = Math.abs(endPoint.x - startPoint.x);

int ovalHeight = Math.abs(endPoint.y - startPoint.y);

g2d.drawOval(ovalX, ovalY, ovalWidth, ovalHeight);

break;

}

}

}

};

drawingPanel.setBackground(Color.WHITE);

add(drawingPanel, BorderLayout.CENTER);

drawingPanel.addMouseListener(new MouseAdapter() {

@Override

public void mousePressed(MouseEvent e) {

startPoint = e.getPoint();

}

@Override

public void mouseReleased(MouseEvent e) {

endPoint = e.getPoint();

drawingPanel.repaint();

}

});

lineButton.addActionListener(e -> selectedShape = "Line");

rectangleButton.addActionListener(e -> selectedShape = "Rectangle");

ovalButton.addActionListener(e -> selectedShape = "Oval");

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

Week12a frame = new Week12a();

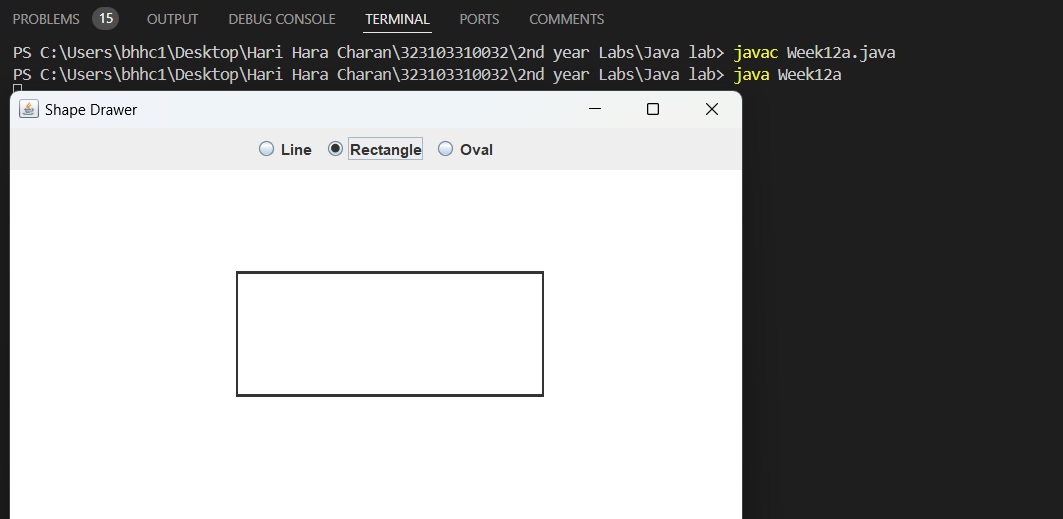
frame.setVisible(true);

});

}

}

***Output:***



**(B) AIM:** To write a program to create a Table inside a JFrame.

***Program:***

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class Week12b {

public static void main(String[] args) {

JFrame frame = new JFrame("Table Example");

frame.setSize(600, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

String[] columnNames = {"SNO", "Name", "Age", "Department", "Salary"};

Object[][] data = {

{1, "Hari", 51, "CSE", 150000},

{2, "Praveen", 19, "CSE", 60000},

{3, "Akhil", 35, "CSE", 70000},

{4, "Bharat", 11, "NONE", 800}

};

JTable table = new JTable(new DefaultTableModel(data, columnNames));

JScrollPane scrollPane = new JScrollPane(table);

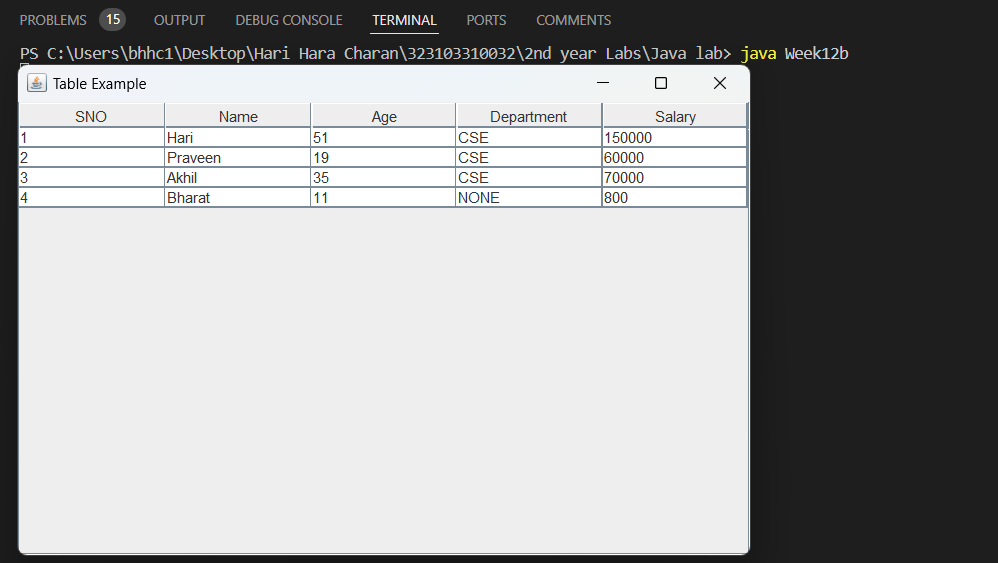
frame.add(scrollPane);

frame.setVisible(true);

}

}

***Output:***



**(C) AIM:** To Create an interface that illustrates JFileChooser class and read CSV file containing employee data of various departments and display the records department wise on the interface.

***Program:***

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.\*;

import java.util.ArrayList;

import java.util.List;

public class Week12c extends JFrame {

private JTable table;

private JComboBox<String> departmentComboBox;

private DefaultTableModel tableModel;

private List<String[]> employeeData;

public Week12c() {

setTitle("Employee Data Viewer");

setSize(800, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel controlPanel = new JPanel();

JButton loadFileButton = new JButton("Load CSV File");

departmentComboBox = new JComboBox<>(new String[]{"All Departments"});

controlPanel.add(loadFileButton);

controlPanel.add(new JLabel("Filter by Department:"));

controlPanel.add(departmentComboBox);

add(controlPanel, BorderLayout.NORTH);

tableModel = new DefaultTableModel(new String[]{"ID", "Name", "Age", "Department", "Salary"}, 0);

table = new JTable(tableModel);

add(new JScrollPane(table), BorderLayout.CENTER);

employeeData = new ArrayList<>();

loadFileButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

loadCSVFile();

}

});

departmentComboBox.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

filterByDepartment((String) departmentComboBox.getSelectedItem());

}

});

}

private void loadCSVFile() {

JFileChooser fileChooser = new JFileChooser();

int result = fileChooser.showOpenDialog(this);

if (result == JFileChooser.APPROVE\_OPTION) {

File selectedFile = fileChooser.getSelectedFile();

readCSVFile(selectedFile);

}

}

private void readCSVFile(File file) {

employeeData.clear();

tableModel.setRowCount(0);

try (BufferedReader br = new BufferedReader(new FileReader(file))) {

String line;

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

String[] data = line.split(",");

employeeData.add(data);

}

populateDepartmentComboBox();

filterByDepartment("All Departments");

} catch (IOException e) {

JOptionPane.showMessageDialog(this, "Error reading the file: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void populateDepartmentComboBox() {

departmentComboBox.removeAllItems();

departmentComboBox.addItem("All Departments");

employeeData.stream()

.map(data -> data[3])

.distinct()

.forEach(departmentComboBox::addItem);

}

private void filterByDepartment(String department) {

tableModel.setRowCount(0);

for (String[] data : employeeData) {

if ("All Departments".equals(department) || data[3].equals(department)) {

tableModel.addRow(data);

}

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

Week12c frame = new Week12c();

frame.setVisible(true);

});

}

}

***Output:***

