TERMWORK 1.A

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

1.1) It is required to store and analyze data about 6 car manufacturer’s sales data in all the 12 months of a year. Demonstrate how you would store the data in a two dimensional matrix and do the following

1. Write a function to Find for a given car manufacturer, the month in which, maximum no. of cars are sold.
2. Write a function to Find the average number of cars sold for each car manufacturer
3. Write a function to Find the total number of cars sold for each car manufacturer.
4. Assume – Row index 0 - ‘Maruti Suzuki’, 1 – ‘Hyundai’ 2 – ‘Tata Motors’ 3-‘KIA’ 4 – ‘BMW’ 5 – ‘Renault’
5. Col index 0 –‘Jan’, 1-‘Feb’………………………………….11 –‘Dec’

SOURCE CODE

*package* termwork1;  
  
*import* java.util.\*;  
*import* java.lang.\*;  
  
*public class* termwork11 {  
 *static* Scanner read = *new* Scanner(System.in);  
 *static* Random rand = *new* Random();  
  
 *static void* findMax(*int*[][] a) {  
 *int*[] largest = *new int*[6];  
 *int*[] n = *new int*[6];  
 *for* (*int* i = 0; i < 6; i++) {  
 *int* m = 0;  
 *for* (*int* j = 0; j < 12; j++) {  
 *if* (m < a[i][j]) {  
 m = a[i][j];  
 n[i] = j;  
 }  
 }  
 largest[i] = m;  
 }  
 System.out.println("\nthe maximum number of sales made is");  
 *for* (*int* i = 0; i < 6; i++) {  
 System.out.println("The best sale by " + i + " is " + largest[i] + " which was made in the month " + n[i]);  
 }  
 }  
  
 *static void* averageSale(*int*[][] a) {  
 *int*[] average = *new int*[6];  
 *int* av;  
 *for* (*int* i = 0; i < 6; i++) {  
 av = 0;  
 *for* (*int* j = 0; j < 12; j++) {  
 av = av + a[i][j];  
 }  
 av = av / 12;  
 average[i] = av;  
 }  
 *for* (*int* i = 0; i < 6; i++) {  
 System.out.println("average sales made by company " + i + " is " + average[i]);  
 }  
 }  
  
 *static void* totalSale(*int*[][] a) {  
 *int*[] average = *new int*[6];  
 *int* av;  
 *for* (*int* i = 0; i < 6; i++) {  
 av = 0;  
 *for* (*int* j = 0; j < 12; j++) {  
 av = av + a[i][j];  
 }  
 average[i] = av;  
 }  
 *for* (*int* i = 0; i < 6; i++) {  
 System.out.println("Total sales made by company " + i + " is " + average[i]);  
 }  
 }  
  
 *public static void* main(String[] args) {  
*// int c = 6, m = 12;  
 int*[][] data = *new int*[6][12];  
 *for* (*int* i = 0; i < 6; i++) {  
 System.out.println("\n enter the data for the " + i + " manufracturer ");  
 *for* (*int* j = 0; j < 12; j++) {  
 data[i][j] = read.nextInt());  
 System.out.print(data[i][j]+" ");  
 }  
 }  
 findMax(data);  
 averageSale(data);  
 totalSale(data);  
 }  
}

**OUTPUT:**

enter the data for the 0 manufracturer

58 35 58 29 6 83 70 40 20 74 74 65

enter the data for the 1 manufracturer

98 23 69 91 61 14 91 15 89 8 3 5

enter the data for the 2 manufracturer

85 89 17 73 68 86 39 80 58 5 71 4

enter the data for the 3 manufracturer

76 86 88 26 71 90 85 52 77 68 18 24

enter the data for the 4 manufracturer

87 90 18 3 89 14 70 32 90 65 64 57

enter the data for the 5 manufracturer

68 41 52 33 83 92 24 87 69 56 66 53

the maximum number of sales made is

The best sale by 0 is 83 which was made in the month 5

The best sale by 1 is 98 which was made in the month 0

The best sale by 2 is 89 which was made in the month 1

The best sale by 3 is 90 which was made in the month 5

The best sale by 4 is 90 which was made in the month 1

The best sale by 5 is 92 which was made in the month 5

average sales made by company 0 is 51

average sales made by company 1 is 47

average sales made by company 2 is 56

average sales made by company 3 is 63

average sales made by company 4 is 56

average sales made by company 5 is 60

Total sales made by company 0 is 612

Total sales made by company 1 is 567

Total sales made by company 2 is 675

Total sales made by company 3 is 761

Total sales made by company 4 is 679

Total sales made by company 5 is 724

TERMWORK 1B

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

1.4) A company has 10 zonal sales offices in four zones namely, North, East, West and South. The company wants to organize the sales data of each of the office in each zone and find answers to queries such as,

1. Which office has performed the highest sales in each zone?
2. What is the average sales done by all the offices in each zone?
3. Which office among each zone is the poorly office?

You are required to answer the following:

1. How do you organize the above data?
2. How do you provide answers to the above queries?

Design a Java application for the same and demonstrate the correctness of the solution

SOURCE CODE:

*import* java.util.Scanner;  
*import* java.util.Random;  
  
*public class* termwork2busingarrays2d {  
  
 *public static void* main(String[] args) {  
 Scanner read = *new* Scanner(System.in);  
 Random rand = *new* Random();  
 *int* data[][] = *new int*[4][10];  
 *int* p, i, j;  
 *int* bestOffice[] = *new int*[4];  
 *int* worstOffice[] = *new int*[4];  
 *int* maxs = 0, mins = 9999;  
 *int* max[] = *new int*[4];  
 *int* min[] = *new int*[4];  
 String zone[] = {"North", "South", "East", "West"};  
 *float* average[] = *new float*[4];  
 *for* (i = 0; i < 4; i++) {  
 System.out.println("The zone is: " + zone[i]);  
 *float* sum = 0;  
 maxs = 0;  
 mins = 9999;  
 *for* (j = 0; j < 10; j++) {  
 System.out.println("enter the sales made by office " + j + " :");  
*// data[i][j] = read.nextInt();* data[i][j] = rand.nextInt(20, 80);  
 System.out.print(data[i][j] + "\n");  
 sum = sum + data[i][j];  
  
 *if* (data[i][j] > maxs ) {  
 maxs = data[i][j];  
 bestOffice[i] = j;  
 }  
 *if* (mins > data[i][j]) {  
 mins = data[i][j];  
 worstOffice[i] = j;  
 }  
  
 }  
 average[i] = sum / 10;  
 max[i] = maxs;  
 min[i] = mins;  
 }  
 *for* (i = 0; i < 4; i++) {  
 System.out.println("\nThe best office in the " + zone[i] + " zone is:");  
 System.out.print(bestOffice[i]);  
 System.out.println("\n this office made " + max[i] + " sales");  
 }  
 System.out.println("\n");  
 *for* (i = 0; i < 4; i++) {  
 System.out.println("\nThe worst office in the " + zone[i] + " zone is:");  
 System.out.print(worstOffice[i]);  
 System.out.println("\nthis office made " + min[i] + " sales");  
 }  
 System.out.println("\n");  
 *for* (i = 0; i < 4; i++) {  
 System.out.println("\nThe average sales in the " + zone[i] + " zone is :");  
 System.out.print(average[i]);  
 }  
 System.out.println("\n");  
 }  
}

OUTPUT:

The zone is: North

enter the sales made by offices :

77 71 42 46 50 31 62 22 42 20

The zone is: South

enter the sales made by offices :

73 43 59 79 21 54 55 25 59 23

The zone is: East

enter the sales made by offices :

35 67 65 35 26 73 36 21 55 38

The zone is: West

enter the sales made by offices :

70 37 50 37 41 41 45 43 68 66

The best office in the North zone is:

0

this office made 77 sales

The best office in the South zone is:

3

this office made 79 sales

The best office in the East zone is:

5

this office made 73 sales

The best office in the West zone is:

0

this office made 70 sales

The worst office in the North zone is:

9

this office made 20 sales

The worst office in the South zone is:

4

this office made 21 sales

The worst office in the East zone is:

7

this office made 21 sales

The worst office in the West zone is:

1

this office made 37 sales

The average sales in the North zone is :

46.3

The average sales in the South zone is :

49.1

The average sales in the East zone is :

45.1

The average sales in the West zone is :

49.8

TERMWORK 2A

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

2.1) Design a class by name myCircle to model Circle geometrical object with its center and radius that enables:

1. Initializing the center, radius and
2. Compute area, perimeter, and diameter of the circle object/s.

TASK 1: Identify member variable/s and their types

TASK 2: Identify Constructor/s along with their arguments (if any) to initialize the member variables

TASK 3: Identify the methods along with their arguments and return types.

SOURCE CODE

*package* termwork2;  
*import* java.util.\*;  
  
*class* myCircle{  
 *int* r;  
 *int* x=0,y=0;  
 myCircle(*int* r){  
 *this*.r=r;  
 }  
 *void* area(){  
 *double* area;  
 area= (3.14\*r\*r);  
 System.out.println("the area is "+area);  
 }  
 *void* diameter(){  
 System.out.println("the diameter is "+(r\*2));  
 }  
 *void* perimeter(){  
 System.out.println("the perimeter of the circle is "+2\*3.14\*r);  
 }  
}  
*public class* termwork2a {  
 *static* Scanner read=*new* Scanner(System.in);  
 *public static void* main(String[] args) {  
 myCircle c1=*new* myCircle(3);  
 c1.area();  
 c1.diameter();  
 c1.perimeter();  
 }  
}

OUTPUT CODE:

the area is 28.259999999999998

the diameter is 6

the perimeter of the circle is 18.84

Process finished with exit code 0

TERMWORK 2B

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

*package* termwork2;  
  
*import* java.util.Scanner;  
  
*class* Lamp {  
 *boolean* status;  
  
 *void* turnOn() {  
 System.out.println("⚫");  
 status = *true*;  
  
 }  
  
 *void* turnoff() {  
 System.out.println("◯");  
 status = *false*;  
 }  
  
 *void* state() {  
 *if* (status)  
 System.out.println("light is on ⚫ ");  
 *else* System.out.println("Light is off ◯");  
 }  
}  
  
*public class* termwork2b {  
 *static* Scanner read = *new* Scanner(System.in);  
  
 *public static void* main(String[] args) {  
 Lamp l1 = *new* Lamp();  
 *while* (*true*) {  
 System.out.println("1.lightON 2.lightOFF 3.Check 4.close");  
 *int* choice = read.nextInt();  
 *int* br = 0;  
 *switch* (choice) {  
 *case* 1 -> l1.turnOn();  
 *case* 2 -> l1.turnoff();  
 *case* 3 -> l1.state();  
 *case* 4 -> br = 1;  
 }  
 *if* (br == 1) {  
 *break*;  
 }  
 }  
 }  
}

OUTPUT:

1.lightON 2.lightOFF 3.Check 4.close

1

⚫

1.lightON 2.lightOFF 3.Check 4.close

2

◯

1.lightON 2.lightOFF 3.Check 4.close

1

⚫

1.lightON 2.lightOFF 3.Check 4.close

3

light is on ⚫

1.lightON 2.lightOFF 3.Check 4.close

4

TERMWORK 3A

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

3.2) Define a class to represent a rectangle in which constructors and parameterized methods are used. It also has a method to compute area of rectangle.

1. First make a class rectangle in which we declare the parameterized constructor.
2. Then demonstrate the use of parameterized method.
3. Use a method to compute area of rectangle.

Create a class to demonstrate the call of the methods in previously created class rectangle holding constructors, parameterized methods and method to compute area of rectangle

*package* termwork3;  
*class* rectangle{  
 *int* len,br;  
 rectangle(){  
 len=0;  
 br=0;  
 }  
 rectangle(*int* len,*int* br){  
 *this*.len=len;  
 *this*.br=br;  
 }  
 *void* area(){  
 System.out.println("the area of the rectangle is "+ len\*br);  
 }  
  
}  
*public class* termwork3a {  
 *public static void* main(String[] args) {  
 rectangle r1=*new* rectangle(5,5);  
 r1.area();  
 }  
}

OUTPUT:

the area of the rectangle is 25

TERMWORK 3B

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

3.3) Write a Java program to represent a Complex number. Include member functions to:

1. Initialize a complex number to a default value of zero (default constructor)
2. Initialize a complex number to a user defined value (parameterized constructor)
3. Add two complex numbers and return the result. (Parameterized method)
4. Subtract two complex numbers and return the result. (Parameterized method)
5. Display a complex number. (non-parameterized method)
6. *package* termwork3;  
   *class* complex{  
    *int* r,i;  
    complex(){  
    *this*.i=0;  
    *this*.r=0;  
    }  
    complex(*int* r, *int* i){  
    *this*.i=i;  
    *this*.r=r;  
    }  
    complex add(complex c1,complex c2){  
    complex temp=*new* complex();  
    temp.r=c1.r+c2.r;  
    temp.i=c1.i+c2.i;  
    *return* temp;  
    }  
    complex sub(complex c1,complex c2){  
    complex temp=*new* complex();  
    temp.r=c1.r-c2.r;  
    temp.i=c1.i-c2.i;  
    *return* temp;  
    }  
    *void* display(){  
    System.out.println("the real part is"+r);  
    System.out.println("the imaginary part is"+i);  
    }  
   }  
   *public class* termwork3b {  
    *public static void* main(String[] args) {  
    complex c1 =*new* complex(4,5);  
    complex c2 =*new* complex(4,5);  
    complex result=*new* complex(0,0);  
    result=result.add(c1,c2);  
    result.display();  
    }  
   }

OUTPUT: 8 + 10i

TERMWORK 5A

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

* 1. Implement a linear search function by using method overloading concept for an array of integers, double and character elements.

1. *package* termwork5;  
     
   *import* java.util.Scanner;  
   *import* java.util.Arrays;  
     
   *class* MyLinearSearch {  
    *int* linearSearch(*int*[] c, *int* key2) {  
    *for* (*int* i = 0; i < c.length; i++) {  
    *if* (c[i] == key2) {  
    *return* i;  
    }  
    }  
    *return* -1;  
    }  
     
    *double* linearSearch(*double*[] a, *double* key) {  
    *for* (*int* i = 0; i < a.length; i++) {  
    *if* (a[i] == key) {  
    *return* i;  
    }  
    }  
    *return* -1;  
    }  
     
    *int* linearSearch(*char*[] a, *char* key) {  
    *for* (*int* i = 0; i < a.length; i++) {  
    *if* (a[i] == key) {  
    *return* i;  
    }  
    }  
    *return* -1;  
    }  
   }  
     
   *class* terworkb {  
    *public static void* main(String[] args) {  
    MyLinearSearch m = *new* MyLinearSearch();  
    Scanner in = *new* Scanner(System.in);  
    System.out.println("Enter the type of element : ");  
    System.out.println("1:integer \t 2:double \t 3:char");  
    *int* choice = in.nextInt();  
    System.out.println("Enter the number of elements : ");  
    *int* n = in.nextInt();  
    System.out.println("Enter " + n + " elements : ");  
    *switch* (choice) {  
    *case* 1 -> {  
    *int*[] a = *new int*[n];  
    *for* (*int* i = 0; i < n; i++) {  
    a[i] = in.nextInt();  
    }  
    System.out.println("Enter the element to be searched : ");  
    *int* key = in.nextInt();  
    *int* position = m.linearSearch(a, key);  
    check(position);  
    }  
    *case* 2 -> {  
    *double*[] a1 = *new double*[n];  
    *for* (*int* i = 0; i < n; i++) {  
    a1[i] = in.nextDouble();  
    }  
    System.out.println("Enter the element to be searched : ");  
    *double* key1 = in.nextInt();  
    *int* position1 = (*int*) m.linearSearch(a1, key1);  
    check(position1);  
    }  
    *case* 3 -> {  
    *char*[] c = *new char*[n];  
    *for* (*int* i = 0; i < n; i++) {  
    c[i] = in.next().charAt(0);  
    }  
    System.out.println("Enter the element to be searched : ");  
    *char* key2 = in.next().charAt(0);  
    *int* position2 = (*int*) m.linearSearch(c, key2);  
    check(position2);  
    }  
    }  
    }  
     
    *public static void* check(*int* position1) {  
    *if* (position1 >= 0)  
    System.out.println("Element is found at index " + (position1 + 1));  
    *else* System.out.println("Element not found\n");  
    }  
   }

OUTPUT

CASE 1:

Enter the type of element :

1:integer 2:double 3:char

1

Enter the number of elements :

5

Enter 5 elements :

1 2 3 4 5

Enter the element to be searched :

5

Element is found at index 5

CASE 2:

Enter the type of element :

1:integer 2:double 3:char

3

Enter the number of elements :

5

Enter 5 elements :

A E I O U

Enter the element to be searched :

U

Element is found at index 5

TERMWORK 5B

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

* 1. Design a Vehicle class to include as data members, the vehicle’s initial velocity (u), final velocity (v) and acceleration (a). The class must a parameterized constructor to initialize these data members.

Design a class LawsofMotion and write two overloaded methods, computeDistanceTravelled and computeAcceleration to return the distance travelled by the vehicle and acceleration attained respectively, given the different parameter values. The class must have appropriate data members for the following methods to work.

computeDistanceTravelled(float t)- computes and returns the distance travelled as: u\*t+0.5f\*a\*t\*t

computeDistanceTravelled() - computes and returns the distance travelled as (( v\*v –u\*u)/(2\*a));

computeAcceleration(float mass, float force) - computes and returns the acceleration as a=mass/force;

computeAcceleration(float mass, float v, float u,float t) - computes and returns the acceleration as (m\*v-m\*u)/t;

Demonstrate the working by instantiating objects of the above class and verify the working of the overloaded methods.

*package* termwork5;  
*class* Vehicle {  
 *float* u, v, a;  
 *public* Vehicle() {  
 v = 0;  
 u = 0;  
 a = 0;  
 }  
 Vehicle(*float* u, *float* v, *float* a) {  
 *this*.u = u;  
 *this*.v = v;  
 *this*.a = a;  
 }  
}  
*class* LawsofMotion *extends* Vehicle {  
 Vehicle V1 = *new* Vehicle();  
 *float* distance;  
 *float* a;  
 LawsofMotion(*float* u, *float* v, *float* a) {  
 *this*.V1.u = u;  
 *this*.V1.v = v;  
 *this*.V1.a = a;  
 }  
 *void* computeDistanceTravelled(*float* t) {  
 distance = V1.u \* t + 0.5f \* V1.a \* t \* t;  
 System.out.println("The distance travelled is " + distance);  
 }  
 *void* computeDistanceTravelled() {  
 distance = (V1.u \* V1.u) / (2 \* V1.a);  
 System.out.println("The distance travelled is " + distance);  
 }  
 *void* computeAcceleration(*float* mass, *float* force) {  
 a = (mass / force);  
 System.out.println("The acceleration is " + a);  
 }  
 *void* computeAcceleration(*float* mass, *float* v, *float* u, *float* t) {  
 a = (mass \* v - mass \* u) / t;  
 System.out.println("The acceleration is " + a);  
 }  
}  
*public class* termwork5a {  
 *public static void* main(String[] args) {  
 LawsofMotion V = *new* LawsofMotion(4, 5, 6);  
 V.computeDistanceTravelled();  
 V.computeDistanceTravelled(2);  
 V.computeAcceleration(30, 50);  
 V.computeAcceleration(5,10,5,10);  
 }  
}

OUTPUT:

The distance travelled is 1.3333334

The distance travelled is 20.0

The acceleration is 0.6

The acceleration is 2.5