**Practical-1**

**Aim: - Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.**

**Code: -**

class Main

{

public static void main(String args[])

{

System.out.print("Welcome to Java, Learning Java Now and Programming is fun..");

}

}

**Output: -**

****

**Practical-2**

**Aim: - Write a program that solves the following equation and displays the value x and y: 1) 3.4 x + 50.2 y = 44.5 2) 2.1 x + 0.55 y = 5.9(Assume Cramer’ s rule to solve equation ax + by = e x = ed - bf / ad - bc, cx + dy = f y = af - ec / ad - bc).**

**Code: -**

import java.util.Scanner;

class cramer

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Values from Equestion - 1 :");

System.out.print("Enter value of a : ");

double a = sc.nextDouble();

System.out.print("Enter value of b : ");

double b = sc.nextDouble();

System.out.print("Enter value of e : ");

double e = sc.nextDouble();

System.out.println("Values from Equestion - 2 :");

System.out.print("Enter value of c : ");

double c = sc.nextDouble();

System.out.print("Enter value of d : ");

double d = sc.nextDouble();

System.out.print("Enter value of f : ");

double f = sc.nextDouble();

double x = ((e \* d) - (b \* f)) / ((a \* d) - (b \* c));

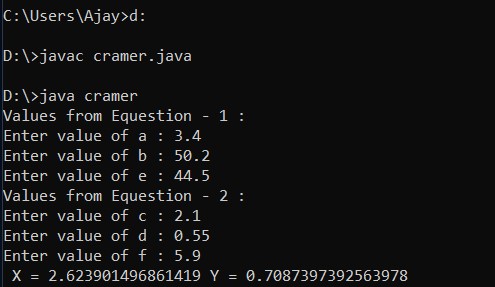
double y = ((a \* f) - (e \* c)) / ((a \* d) - (b \* c));

System.out.print(" X = " + x + " Y = " + y);

}

}

**Output: -**

****

**Practical-3**

**Aim: - Write a program that reads a number in meters, converts it to feet, and displays the result.**

**Code: -**

import java.util.Scanner;

class measure

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter Value in Meters :");

double met = sc.nextDouble();

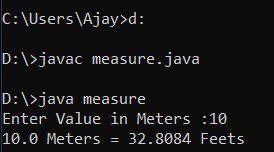
double feet = met \* 3.28084;

System.out.print(met + " Meters = " + feet + " Feets");

}

}

**Output: -**

****

**Practical-4**

**Aim: - Body Mass Index(BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note: 1 pound = 0.45359237 Kg and 1 inch = 0.0254 meters.**

**Code: -**

import java.util.Scanner;

class bmi

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter Your weight in Pound :");

double pou = sc.nextDouble();

System.out.print("Enter Your Height in Inch :");

double inch = sc.nextDouble();

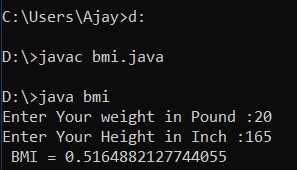
double BMI = (pou \* 0.45359237) / ((inch \* 0.0254) \* (inch \* 0.0254));

System.out.print(" BMI = " + BMI);

}

}

**Output: -**

****

**Practical-5**

**Aim: - Write a program that prompts the user to enter three integers and display the integers in decreasing order.**

**Code: -**

import java.util.Scanner;

class decrease

{

public static void main(String args[])

{

int temp;

Scanner input = new Scanner(System.in);

System.out.print("Enter First Integer :");

int a = input.nextInt();

System.out.print("Enter Second Integer :");

int b = input.nextInt();

if (a < b)

{

temp = a;

a = b;

b = temp;

}

System.out.print("Enter Third Integer :");

int c = input.nextInt();

if (c > b)

{

if (c > a)

{

temp = c;

c = b;

b = a;

a = temp;

}

else

{

temp = c;

c = b;

b = temp;

}

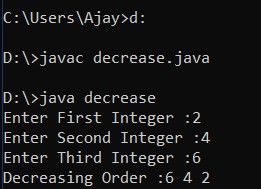
}

System.out.print("Decreasing Order :" + a + " " + b + " " + c);

}

}

**Output: -**

****

**Practical-6**

**Aim: - Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant.**

**Code: -**

import java.util.Scanner;

class aeiou

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter Character : ");

char ch = sc.next().charAt(0);

if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')

{

System.out.print("character is vowel");

}

else

{

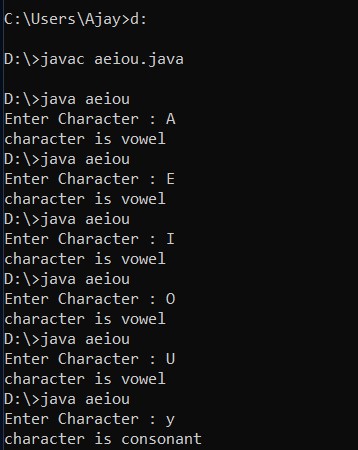
System.out.print("character is consonant");

}

}

}

**Output: -**

****

**Practical-7**

**Aim: - Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.**

**Code: -**

class plate

{

public static void main(String args[])

{

int alpha1 = 'A' + (int)(Math.random() \* ('Z' - 'A'));

int alpha2 = 'A' + (int)(Math.random() \* ('Z' - 'A'));

int alpha3 = 'A' + (int)(Math.random() \* ('Z' - 'A'));

int digit1 = (int)(Math.random() \* 10);

int digit2 = (int)(Math.random() \* 10);

int digit3 = (int)(Math.random() \* 10);

int digit4 = (int)(Math.random() \* 10);

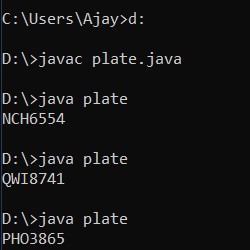
System.out.println("" + (char)(alpha1) + ((char)(alpha2)) +

((char)(alpha3)) + digit1 + digit2 + digit3 + digit4);

}

}

**Output: -**

****

**Practical-8**

**Aim: - Write a program that reads an integer and displays all its smallest factors in increasing order.For example if input number is 120, the output should be   
as follows: 2, 2, 2, 3, 5.**

**Code: -**

import java.util.Scanner;

class fact{

public static void main(String args[])

{

int i= 2;

Scanner sc = new Scanner(System.in);

System.out.print("Enter Integer Value : ");

int number = sc.nextInt();

while (number > 1)

{

if (number % i == 0)

{

System.out.println("Factors: "+i);

number = number / i;

}

else

{

i++;

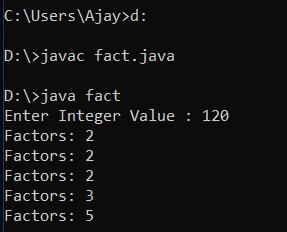
}

}

}

}

**Output: -**

****

**Practical-9**

**Aim: - Write a method with the following method header. Public static int gcd(int num1, int num2) Write a program that prompts the user to enter two integers and compute the gcd of two integers.**

**Code: -**

import java.util.Scanner;

class gcd

{

public static int gcd(int num1,int num2)

{

while(num1!=num2)

{

if(num1>num2)

{

num1=num1-num2;

}

else

{

num2=num2-num1;

}

}

return num1;

}

public static void main(String args[])

{

Scanner S=new Scanner(System.in);

System.out.println("Enter First Integer:");

int num1=S.nextInt();

System.out.println("Enter Second Integer:");

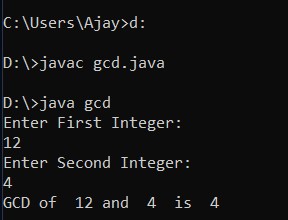
int num2=S.nextInt();

System.out.println("GCD of " +num1 + " and " +num2 + " is " +gcd(num1,num2));

}

}

**Output: -**

****

**Practical-10**

**Aim: - Write a test program that prompts the user to enter ten numbers, invoke   
method to reverse the numbers, display the numbers.**

**Code: -**

import java.util.Scanner;

class reverse

{

public static void reverse(int numbers[])

{

int j=0,temp;

while(j<=numbers.length/2)

{

temp=numbers[j];

numbers[j]=numbers[numbers.length-1-j];

numbers[numbers.length-1-j]=temp;

j++;

}

}

public static void main(String args[])

{

int i=0;

int num\_array[]=new int[5];

Scanner sc= new Scanner(System.in);

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

{

System.out.println("Enter at Position "+ (i+1) + " : ");

num\_array[i] = sc.nextInt();

}

reverse(num\_array);

System.out.println("After reversing numbers in an Array :");

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

{

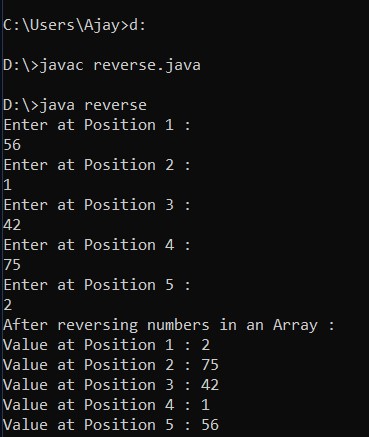
System.out.println("Value at Position "+ (i+1) + " : "+num\_array[i]);

}

}

}

**Output: -**

****

**Practical-11**

**Aim: - Write a program that creates a Random object with seed 1000 and displays the first 100 random integers between 1 and 49 using the NextInt(49)   
method.**

**Code: -**

import java.util.Scanner;

class matrix

{

public static int[][] create\_fill\_matrix()

{

int [][]matrix = new int[6][6];

for(int i=0;i<6;i++)

{

for(int j=0;j<6;j++)

{

matrix[i][j]=(int)((Math.random()\*5)%2);

}

}

return matrix;

}

public static void displayMatrix(int [][]matrix)

{

System.out.print("\nMatrix Values \n");

for(int i=0;i<6;i++)

{

for(int j=0;j<6;j++)

{

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

}

System.out.println();

}

}

public static void main(String[] args)

{

int my\_matrix[][];

int i,j,cnt;

my\_matrix=create\_fill\_matrix();

displayMatrix(my\_matrix);

System.out.println("\nRows Having ODD no of 1s");

for(i=0;i<6;i++)

{

cnt=0;

for(j=0;j<6;j++)

{

if(my\_matrix[i][j]==1)

{

cnt++;

}

}

if(cnt%2!=0)

{

System.out.println("Row - "+(i+1)+" have ODD no of 1s");

}

}

System.out.println("\nColumns Having ODD no of 1s");

for(i=0;i<6;i++)

{

cnt=0;

for(j=0;j<6;j++)

{

if(my\_matrix[j][i]==1)

{

cnt++;

}

}

if(cnt%2!=0)

{

System.out.println("Column - "+(i+1)+" have ODD no of 1s");

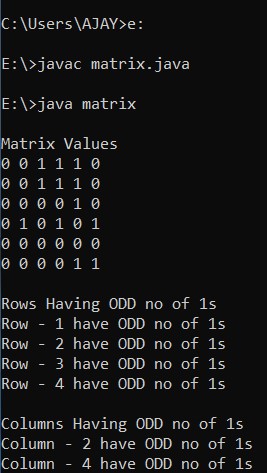
}

}

}

}

**Output: -**



**Practical-12**

**Aim: - Write a program that creates a Random object with seed 1000 and displays the first 100 random integers between 1 and 49 using the NextInt(49) method.**

**Code: -**

import java.util.Random;

class random

{

public static void main(String[] args)

{

Random random = new Random(1000);

for (int i=0;i<100;i++)

{

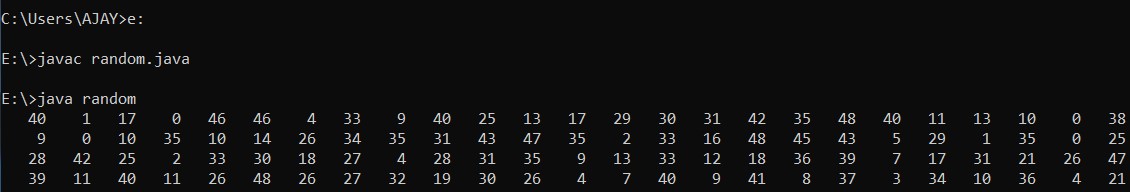
System.out.format("%5d",random.nextInt(49));

}

}

}

**Output: -**

****

**Practical-13**

**Aim: - Write a program for calculator to accept an expression as a string in which the operands and operator are separated by zero or more spaces. For ex: 3+4 and 3 + 4 are acceptable expressions.**

**Code: -**

import java.util.Scanner;

class express

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter Equation:");

String str= sc.nextLine();

String a=str.replaceAll(" ","");

if(a.length()<3)

{

System.out.println("Minimum 2 Opearator and 1 Opearand Required..");

System.exit(0);

}

int result=0;

int i=0;

while(a.charAt(i)!='+' && a.charAt(i)!='-' && a.charAt(i)!='\*' && a.charAt(i)!='/')

{

i++;

}

switch(a.charAt(i))

{

case '+':

result = Integer.parseInt(a.substring(0,i))+Integer.parseInt(a.substring(i+1,a.length()));

break;

case '-':

result = Integer.parseInt(a.substring(0,i))-Integer.parseInt(a.substring(i+1,a.length()));

break;

case '\*':

result = Integer.parseInt(a.substring(0,i))\*Integer.parseInt(a.substring(i+1,a.length());

break;

case '/':

result = Integer.parseInt(a.substring(0,i))/Integer.parseInt(a.substring(i+1,a.length());

break;

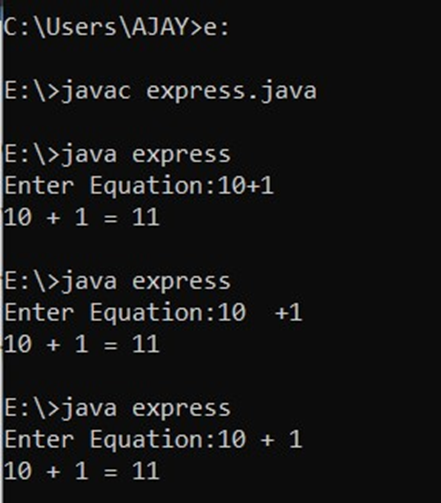
}

System.out.println(a.substring(0,i) + ' '+a.charAt(i)+' ' + a.substring(i+1,a.length())+ " = " + result);

}

}

**Output: -**

****

**Practical-14**

**Aim: - Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object’s to String() method.**

**Code: -**

import java.util.ArrayList;

import java.util.Date;

class invoke

{

public static void main(String args[])

{

ArrayList<Object> arr\_list = new ArrayList<Object>();

arr\_list.add(new Loan(5000.50));

arr\_list.add(new Date());

arr\_list.add(new String("String class"));

arr\_list.add(new Circle(3.45));

for (int i = 0; i < arr\_list.size(); i++)

{

System.out.println((arr\_list.get(i)).toString());

}

}

}

class Circle

{

double radius;

Circle(double r)

{

this.radius=r;

}

public String toString()

{

return "Circle with Radius "+this.radius;

}

}

class Loan

{

double amount;

Loan(double amt)

{

this.amount=amt;

}

public String toString()

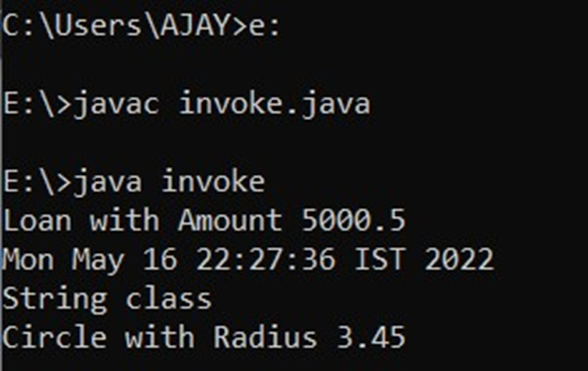
{

return "Loan with Amount "+this.amount;

}

}

**Output :-**



**Practical-15**

**Aim :- Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a  NumberFormatException if the string is not a binary string.**

**Code :-**

import java.util.Scanner;

class bin2dec

{

public static int bin2Dec(String binaryString) throws NumberFormatException

{

int decimal = 0;

int strLength=binaryString.length();

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

{

if (binaryString.charAt(i) < '0' || binaryString.charAt(i) > '1')

{

throw new NumberFormatException("The Input String is not Binary");

}

decimal += (binaryString.charAt(i)-'0') \* Math.pow(2, strLength-1-i);

}

return decimal;

}

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter Binary Value : ");

String str = sc.nextLine();

try

{

System.out.println("Value = " + bin2Dec(str));

}

catch(NumberFormatException e)

{

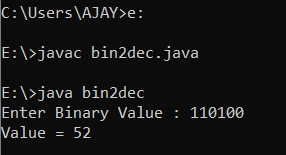
System.out.println(e);

}

}

}

**Output :-**

****

**Practical-16**

**Aim :- Write a program that prompts the user to enter a decimal number and displays the number in a fraction.Hint: Read the decimal number as a string,  extract the integer part and fractional part from the string.**

**Code :-**

import java.util.Scanner;

import java.math.BigInteger;

class biginteger

{

public static void main(String args[])

{

Scanner sc= new Scanner(System.in);

Double d;

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

String[] decimal=sc.nextLine().split("[.]");

BigInteger b1=new BigInteger(decimal[0]);

BigInteger b2=new BigInteger((decimal[1]));

if(decimal[0].charAt(0)=='-')

{

d = b1.doubleValue()-(b2.intValue()/Math.pow(10,decimal[1].length()));

}

else

{

d=b1.doubleValue() + (b2.intValue() / Math.pow(10, decimal[1].length()));

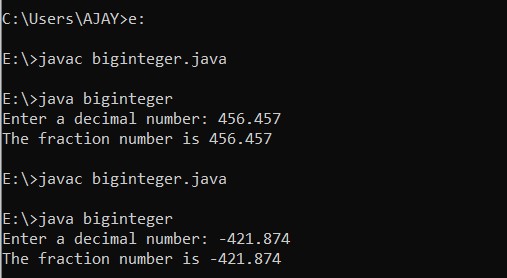
}

System.out.println("The fraction number is "+d);

}

}

**Output :-**

****

**Practical-17**

**Aim :- Write a program that displays a tic-tac-toe board. A cell may be X, O, or empty. What to display at each cell is randomly decided. The X and O are images in the files X.gif and O.gif.**

**Code :-**

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class App extends Application

{

@Override

public void start(Stage primaryStage) throws Exception

{

primaryStage.setTitle("Tic-Tac-Toe by kt");

GridPane gridPane = new GridPane();

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

{

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

{

int n = (int) (Math.random() \* 3);

if (n == 0)

{

gridPane.add(createX(), i, j);

}

else if (n == 1)

{

gridPane.add(createO(), i, j);

}

Else

{

continue;

}

}

}

Scene primaryScene = new Scene(gridPane, 300, 300);

primaryStage.setScene(primaryScene);

primaryStage.show();

}

public VBox createX()

{

Image imageX = new Image("E:/ZIP/x.jpg");

ImageView imageViewX = new ImageView(imageX);

VBox xBox = setProp(imageViewX);

return xBox;

}

public VBox createO()

{

Image imageO = new Image("E:/ZIP/o.jpg");

ImageView imageViewO = new ImageView(imageO);

VBox oBox = setProp(imageViewO);

return oBox;

}

public VBox setProp(ImageView iv)

{

iv.setFitHeight(50);

iv.setFitWidth(50);

iv.setPreserveRatio(true);

VBox vBox = new VBox();

vBox.getChildren().add(iv);

vBox.setStyle("-fx-border-color: orange");

return vBox;

}

public static void main(String args[])

{

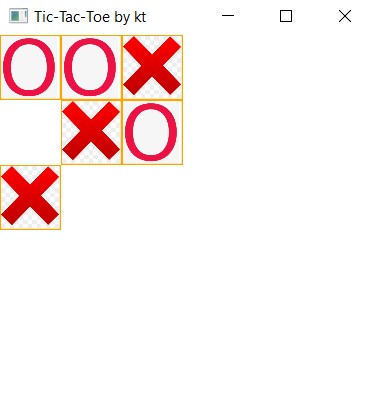
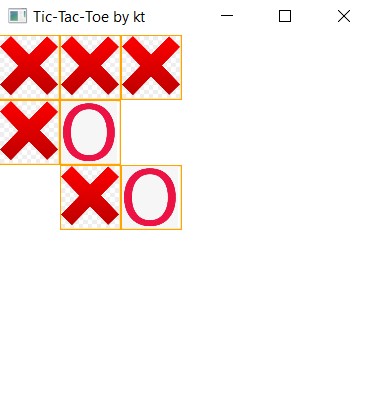
// Here you can work with args - command line parameters

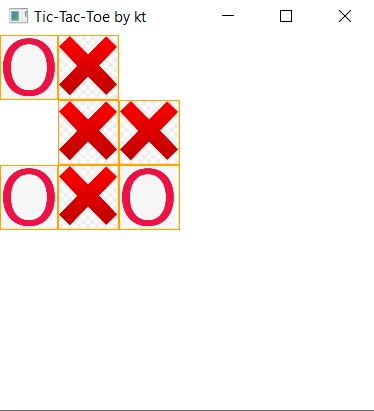
Application.launch(args);

}

}

**Output :-**

**First run :- Second run :-**

**Third run :-**

**Practical-18**

**Aim :- Write a program that** **moves a circle up, down, left or right using arrow keys.**

**Code :-**

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.shape.Circle;

import javafx.scene.layout.Pane;

import javafx.geometry.Insets;

import javafx.stage.Stage;

public class App extends Application

{

@Override

public void start(Stage primaryStage)

{

Pane pane = new Pane();

pane.setPadding(new Insets(30, 30, 30, 30));

Circle circle = new Circle(30, 30, 30);

pane.getChildren().add(circle);

pane.setOnKeyPressed(e -> {

switch (e.getCode())

{

case UP : circle.setCenterY(circle.getCenterY() >

circle.getRadius() ? circle.getCenterY() - 15 :

circle.getCenterY()); break;

case DOWN : circle.setCenterY(circle.getCenterY() <

pane.getHeight() - circle.getRadius() ?

circle.getCenterY() + 15 : circle.getCenterY());

break;

case LEFT : circle.setCenterX(circle.getCenterX() >

circle.getRadius() ? circle.getCenterX() - 15 :

circle.getCenterX()); break;

case RIGHT : circle.setCenterX(circle.getCenterX() <

pane.getWidth() - circle.getRadius() ?

circle.getCenterX() + 15: circle.getCenterX());

default:

break;

}

});

Scene scene = new Scene(pane, 200, 200);

primaryStage.setTitle("App");

primaryStage.setScene(scene);

primaryStage.show();

pane.requestFocus();

}

public static void main(String args[])

{

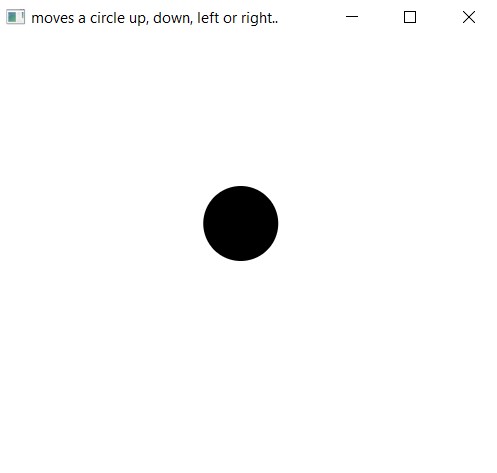
Application.launch(args);

}

}

**Output :-**

Note: Click the arrow keys and moves a circle up, down, left or right.



**Practical-19**

**Aim :- Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.**

**Code :-**

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.layout.StackPane;

import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;

import javafx.stage.Stage;

import java.io.IOException;

public class App extends Application {

@Override

public void start(Stage stage) throws IOException {

StackPane pane = new StackPane();

Scene scene = new Scene(pane, 300, 300);

Circle circle = new Circle(50);

circle.setStroke(Color.RED);

circle.setFill(Color.BLUE);

circle.setStrokeWidth(100);

pane.getChildren().addAll(circle);

scene.setOnMousePressed(e ->

{

circle.setStroke(Color.RED);

});

scene.setOnMouseReleased(e ->

{

circle.setStroke(Color.BLUE);

});

stage.setTitle("Click on the window");

stage.setScene(scene);

stage.show();

}

public static void main(String args[])

{

Application.launch(args);

}

}

**Output :-**

**Note :-** When Mouse Click is Released..



**Note :-** When Mouse Click is Clicked..



**Practical-20**

**Aim :- Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.**

**Code :-**

import javafx.application.Application;

import javafx.stage.Stage;

import javafx.scene.Scene;

import javafx.geometry.Pos;

import javafx.scene.control.Button;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Pane;

import javafx.scene.layout.BorderPane;

import javafx.scene.text.Text;

import javafx.scene.control.RadioButton;

import javafx.scene.control.ToggleGroup;

import javafx.scene.paint.Color;

class App extends Application

{

protected Text text = new Text(50, 50, "CodingKick");

@Override

public void start(Stage primaryStage) {

HBox paneForButtons = new HBox(20);

Button btLeft = new Button("<=");

Button btRight = new Button("=>");

paneForButtons.getChildren().addAll(btLeft, btRight);

paneForButtons.setAlignment(Pos.CENTER);

BorderPane pane = new BorderPane();

pane.setBottom(paneForButtons);

HBox paneForRadioButtons = new HBox(20);

RadioButton rbRed = new RadioButton("Red");

RadioButton rbYellow = new RadioButton("Yellow");

RadioButton rbBlack = new RadioButton("Black");

RadioButton rbOrange = new RadioButton("Orange");

RadioButton rbGreen = new RadioButton("Green");

paneForRadioButtons.getChildren().addAll(rbRed, rbYellow,

rbBlack, rbOrange, rbGreen);

ToggleGroup group = new ToggleGroup();

rbRed.setToggleGroup(group);

rbYellow.setToggleGroup(group);

rbBlack.setToggleGroup(group);

rbOrange.setToggleGroup(group);

rbGreen.setToggleGroup(group);

Pane paneForText = new Pane();

paneForText.setStyle("-fx-border-color: black");

paneForText.getChildren().add(text);

pane.setCenter(paneForText);

pane.setTop(paneForRadioButtons);

btLeft.setOnAction(e -> text.setX(text.getX() - 10));

btRight.setOnAction(e -> text.setX(text.getX() + 10));

rbRed.setOnAction(e -> {

if (rbRed.isSelected()) {

text.setFill(Color.RED);

}

});

rbYellow.setOnAction(e -> {

if (rbYellow.isSelected()) {

text.setFill(Color.YELLOW);

}

});

rbBlack.setOnAction(e -> {

if (rbBlack.isSelected()) {

text.setFill(Color.BLACK);

}

});

rbOrange.setOnAction(e -> {

if (rbOrange.isSelected()) {

text.setFill(Color.ORANGE);

}

});

rbGreen.setOnAction(e -> {

if (rbGreen.isSelected()) {

text.setFill(Color.GREEN);

}

});

Scene scene = new Scene(pane, 450, 150);

primaryStage.setTitle("use the radio button to change the color");

primaryStage.setScene(scene);

primaryStage.show();

}

public static void main(String args[])

{

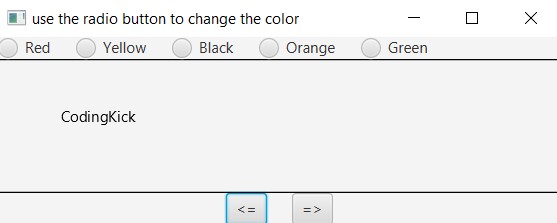
Application.launch(args);

}

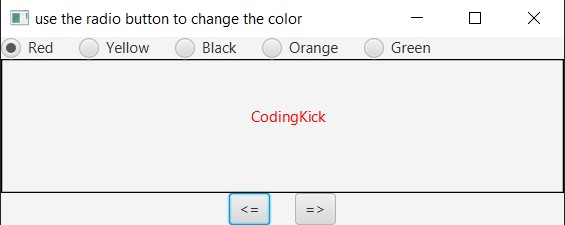
}

**Output :-**

**Note :- Before use button to move the message to the left and right and use the radio button to change the color…**

****

**Note :- After use button to move the message to the left and right and After use the radio button to change the color…**



**Practical-21**

**Aim :- Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.**

**Code :-**

import java.io.\*;

import java.util.Scanner;

class file123

{

public static void main(String[] args)

{

try (

PrintWriter pw = new PrintWriter(new FileOutputStream(new File("123.txt"), true));

) {

for (int i = 0; i < 150; i++)

{

pw.print((int)(Math.random() \* 150) + " ");

}

}

catch (FileNotFoundException fnfe)

{

System.out.println("Cannot create the file.");

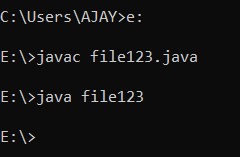
fnfe.printStackTrace();

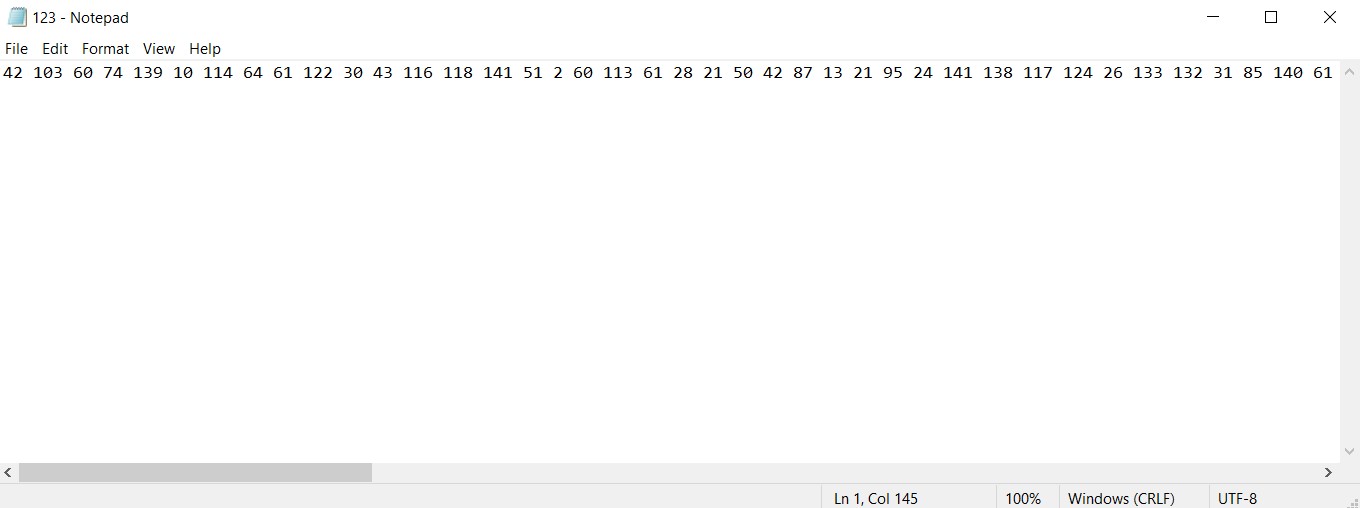
}

}

}

**Output :-**

****

**  
123.txt file :-**

**Practical-22**

**Aim :- Write a recursive method that returns the smallest integer in an array. Write a test program that prompts the user to enter an integer and display its product.**

**Code :-**

import java.util.Scanner;

class prac22

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

int product = 1;

System.out.println("Enter eight integers : ");

int list[] = new int[8];

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

list[i] = input.nextInt();

product = product \* list[i];

}

System.out.println("The smallest element is " + min(list));

System.out.println("The Product of all integers is " + product);

}

// method that finds largest number in an array

public static int min(int list[]) {

int min = list[list.length - 1];

int index = list.length - 1;

return min(list, index, min);

}

// overloaded method

private static int min(int list[], int index, int min) {

if (index < 0)

return min;

else if (list[index] < min)

return min(list, index - 1, list[index]);

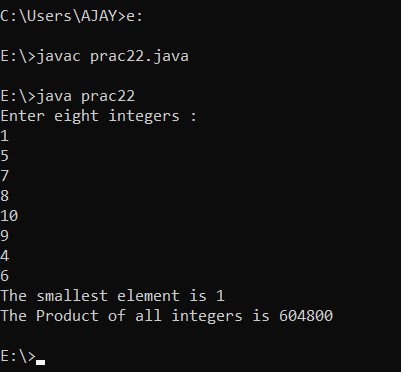
else

return min(list, index - 1, min);

}

}

**Output :-**

****

**Practical-23**

**Aim :- Write a generic method that returns the minimum elements in a two dimensional array.**

**Code** :-

import java.util.Scanner;

class dimension

{

public static void main(String args[])

{

Integer[][] list = new Integer[10][10];

int value = 0;

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

{

for (int j = 0; j < list[i].length; j++)

{

list[i][j] = value++;

}

}

System.out.println("Max = " + max(list));

}

public static <E extends Comparable<E>> E max(E[][] list)

{

E max = list[0][0];

for (E[] elements : list)

{

for (E element : elements)

{

if (element.compareTo(max) > 0)

{

max = element;

}

}

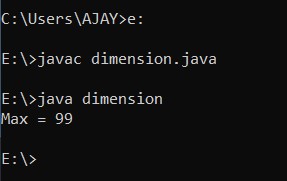
}

return max;

}

}

**Output :-**



**Practical-24**

**Aim :- Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue.**

**Code :-**

**For Class prac24(prac24.java) :-**

class prac24

{

public static void main(String[] args) throws CloneNotSupportedException

{

MyPriorityQueue<Integer> q1 = new MyPriorityQueue<>();

q1.offer(10);

q1.offer(20);

q1.offer(50);

MyPriorityQueue<Integer> q2 = q1.clone();

System.out.print("Queue 1: ");

while (q1.size() > 0) {

System.out.print(q1.remove() + " ");

}

System.out.println();

System.out.print("Queue2: ");

while (q2.size() > 0) {

System.out.print(q2.remove() + " ");

}

}

}

**For Class MyPriorityQueue (MyPriorityQueue.java) :-**

import java.util.PriorityQueue;

class MyPriorityQueue<E> extends PriorityQueue<E> implements Cloneable

{

@Override

public MyPriorityQueue<E> clone() throws CloneNotSupportedException

{

MyPriorityQueue<E> temp = new MyPriorityQueue<>();

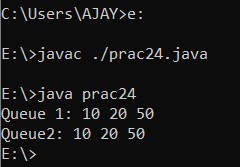
temp.addAll((MyPriorityQueue<E>) super.clone());

return temp;

}

}

**Output :-**

****

**Practical-25**

**Aim :- Write a program that reads words from a text file and displays all the nonduplicate words in descending order.The text file is passed as a command-line argument.**

**Code :-**

import java.io.\*;

import java.util.\*;

class commandline

{

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

File fin = new File(args[0]);

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

StringBuffer buffer = new StringBuffer();

String str;

while ((str = br.readLine()) != null) { // reading the text file

buffer.append(str); // storing text in StringBuffer

buffer.append(" "); // Separating words by spaces

}

ArrayList list = new ArrayList(); // Declaring ArrayList

StringTokenizer st = new StringTokenizer(buffer.toString().toLowerCase());

while (st.hasMoreTokens()) { // creating a list of words

String s = st.nextToken();

list.add(s);

}

HashSet set = new HashSet(list); // it is created to avoid duplicate

List arrayList = new ArrayList(set); // creating list of words

Comparator c = Collections.reverseOrder();

Collections.sort(arrayList, c);

for (Object obj : arrayList) {

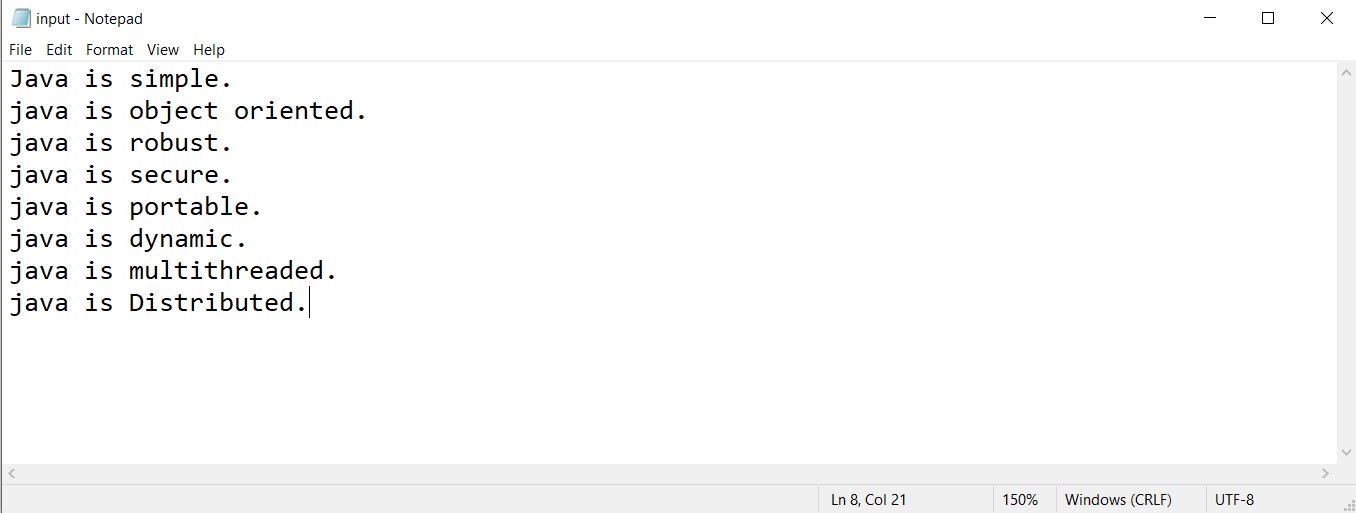
System.out.println(obj.toString()); // displaying content

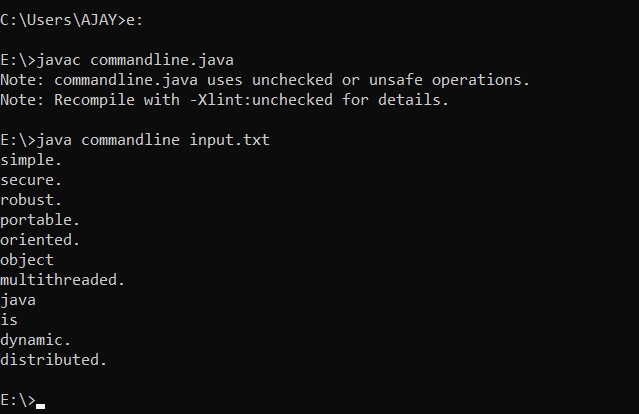
}

}

}

**Input.txt file**

****

**Output :-**