**Name DIV: Roll NoSubject: Java Practical Assignment (403)**

**Index**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No** | **Description** | **Page No.** | **Date Of Submission** | **Sign** |
| 1 | Create STUDENT class having data members roll no and name. Create 5 objects of STUDENT class and take input from the user and print all students’ data in ascending order of name with interval of 1 second |  |  |  |
| 2 | Create an applet which displays a triangle within a circle where the circumference of the circle touches to the all the vertices of the triangle. Provide different colors to both the objects |  |  |  |
| 3 | Write a Java Program that accepts string data. Extract either All Vowels or All Non-Vowels from given Data According to Options Selection. Also Provide an Option to Display Output in Uppercase or Lowercase. |  |  |  |
| 4 | Write an applet that simulates a Rotating wheel. User can start and stop rotation by clicking Start and Stop Button. |  |  |  |
| 5 | Write a Java Program that Accepts String Data from User and then Provide options for Changing case into Any of the Following. (UPPERCASE, lowercase, Sentence case, tOGGLE cASE). |  |  |  |
| 6 | Write an applet that simulates a Digital Clock displaying in the form Hour:Min:Sec. Users should be able to change the color of the Fonts of the clock by the selection provided. |  |  |  |
| 7 | Write a program to draw Smiley face with color using applet |  |  |  |
| 8 | Write a program that accept Book information like Title, Author, Publication and Price for the N book from the user and display books in descending order with interval of 1 second using thread. |  |  |  |
| 9 | Writer a program to demonstrate general operations of circular link list using switch case. |  |  |  |
| 10 | Writer a program to demonstrate general operations of singly link list using switch case. |  |  |  |
| 11 | Write a program which demonstrate sunset using an Applet. |  |  |  |
| 12 | Write a program add/sub/mul/div of 2 numbers in Applet using button. |  |  |  |
| 13 | Accept N number from applet tag and print that many numbers of object using an Applet. |  |  |  |
| 14 | Write a program which demonstrate functionality of creation and use of package. |  |  |  |
| 15 | Accept number from applet tag parameter and draw that many numbers of objects. Fill them with different color. |  |  |  |
|  |  |  |  |  |

**Date:**

**Assign mentor’s Signature:**

**Ms. Rupal Panchal**

1. **Create STUDENT class having data members roll no and name. Create 5 objects of STUDENT class and take input from the user and print all students’ data in ascending order of name with interval of 1 second.**

import java.util.\*;

// create a Student Class.

class STUDENT {

    int rollNo;

    String name;

    public STUDENT(int rollNo, String name) {

        this.rollNo = rollNo;

        this.name = name;

    }

}

public class Assignment\_1 {

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

        Scanner sc = new Scanner(System.in);

        STUDENT[] students = new STUDENT[5];

        // take input from the user and create objects of the STUDENT class

        for (int i = 0; i < 5; i++) {

            System.out.print("Enter roll number of student " + (i + 1) + ": ");

            int rollNo = sc.nextInt();

            System.out.print("Enter name of student " + (i + 1) + ": ");

            String name = sc.next();

            students[i] = new STUDENT(rollNo, name);

        }

        // Ascending  the students by Roll NO.

        for (int i = 0; i < 5; i++) {

            for (int j = 0; j < 4 - i; j++) {

                if (students[j].rollNo>students[j + 1].rollNo){

                    STUDENT temp = students[j];

                    students[j] = students[j + 1];

                    students[j + 1] = temp;

                    //System.out.println(temp );

                }

            }

        }

        // print the  students data with interval of 1 second in ascending order.

        System.out.println("\n Students' data in ascending order: ");

        for (int i = 0; i < 5; i++) {

            System.out.println("Roll number: " + students[i].rollNo + ", Name: " + students[i].name);

            Thread.sleep(1000);

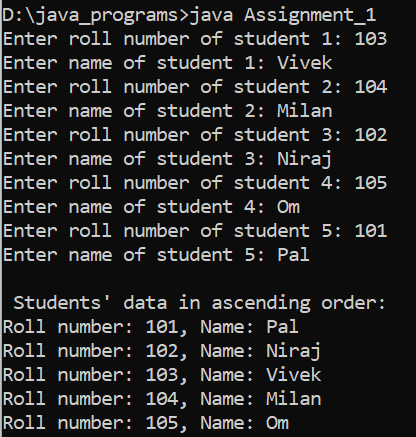
        }

        sc.close();

    }

}

* **Output**:



1. **Create an applet which displays a triangle within a circle where the circumference of the circle touches to the all the vertices of the triangle. Provide different colors to both the objects.**

import java.awt.\*; import java.applet.\*;

/\* <applet code="program2" height=500 width=700> </applet> \*/ public class program2 extends Applet { public void paint(Graphics g) {

g.setColor(Color.green); int x[] = { 350, 150, 550 }; int y[] = { 100, 350, 350 };

int n = 3;

g.fillPolygon(x, y, n);

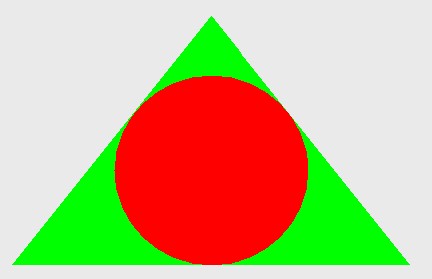
g.setColor(Color.red);

g.fillOval(253, 160, 195, 190);

}

};

* **Output:**



1. **Write a Java Program that accepts string data. Extract either All Vowels or All Non-Vowels from given Data According to Options Selection. Also Provide an Option to Display Output in Uppercase or Lowercase.**

import java.io.\*;

class Letters {

    public void printVovels(String str, int choice) {

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

            if (Character.isAlphabetic(str.charAt(i))) {

                if (str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o'

                        || str.charAt(i) == 'u' || str.charAt(i) == 'A' || str.charAt(i) == 'E' || str.charAt(i) == 'I'

                        || str.charAt(i) == 'O' || str.charAt(i) == 'U') {

                    if (choice == 1) {

                        System.out.print(Character.toUpperCase(str.charAt(i)) + " ");

                    } else {

                        System.out.print(Character.toLowerCase(str.charAt(i)) + " ");

                    }

                }

            }

        }

    }

    public void printNonVovels(String str, int choice) {

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

            if (Character.isAlphabetic(str.charAt(i))) {

                if (str.charAt(i) != 'a' && str.charAt(i) != 'e' && str.charAt(i) != 'i' && str.charAt(i) != 'o'

                        && str.charAt(i) != 'u' && str.charAt(i) != 'A' && str.charAt(i) != 'E' && str.charAt(i) != 'I'

                        && str.charAt(i) != 'O' && str.charAt(i) != 'U') {

                    if (choice == 1) {

                        System.out.print(Character.toUpperCase(str.charAt(i)) + " ");

                    } else {

                        System.out.print(Character.toLowerCase(str.charAt(i)) + " ");

                    }

                }

            }

        }

    }

}

public class Assignment\_3 {

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

        InputStreamReader r = new InputStreamReader(System.in);

        BufferedReader br = new BufferedReader(r);

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

        String str = br.readLine();

        Letters l = new Letters();

        System.out.println("Please enter your choice: ");

        System.out.println("For Vovels: 1");

        System.out.println("For Non vovels: 2");

        String a = br.readLine();

        int choice1 = Integer.parseInt(a);

        System.out.println("Please enter your choice: ");

        System.out.println("For Uppercase: 1");

        System.out.println("For Lowercase: 2");

        String b = br.readLine();

        int choice2 = Integer.parseInt(b);

        if (choice1 == 1) {

            l.printVovels(str, choice2);

        } else if (choice1 == 2) {

            l.printNonVovels(str, choice2);

        } else {

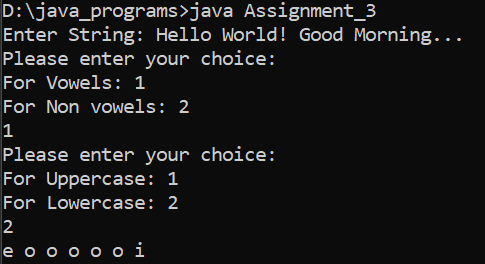
            System.out.println("Please enter valid choice!");

        }

    }

}

* **Output**:



1. **Write an applet that simulates a Rotating wheel. User can start and stop rotation by clicking Start and Stop Button.**

importjava.awt.\*;

importjava.awt.event.\*;

importjavax.swing.\*;

/\*<applet code="RotatingWheel.class" height=300 width=300></applet>\*/

public class RotatingWheel extends JApplet implements ActionListener {

privateJButtonstartButton;

privateJButtonstopButton;

privateWheelComponent wheel;

private Timer timer;

public void init() {

Container content = getContentPane();

content.setLayout(new BorderLayout());

startButton = new JButton("Start");

stopButton = new JButton("Stop");

startButton.addActionListener(this);

stopButton.addActionListener(this);

JPanel control = new JPanel();

control.add(startButton);

control.add(stopButton);

content.add(control, BorderLayout.NORTH);

wheel = new WheelComponent();

content.add(wheel, BorderLayout.CENTER);

timer = new Timer(50, new TimerListener());

timer.setCoalesce(true);

timer.start();

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == startButton) {

timer.start();

} else if (e.getSource() == stopButton) {

timer.stop();

}

}

private class TimerListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

wheel.repaint();

}

}

private class WheelComponent extends JComponent {

private static final int WHEEL\_SIZE = 200;

public void paintComponent(Graphics g) {

int x = getWidth() / 2 - WHEEL\_SIZE / 2;

int y = getHeight() / 2 - WHEEL\_SIZE / 2;

g.setColor(Color.LIGHT\_GRAY);

g.fillOval(x, y, WHEEL\_SIZE, WHEEL\_SIZE);

g.setColor(Color.BLACK);

g.drawOval(x, y, WHEEL\_SIZE, WHEEL\_SIZE);

int offset = (int) (System.currentTimeMillis() / 20) % 360;

g.setColor(Color.RED);

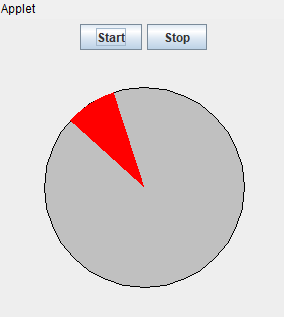
g.fillArc(x, y, WHEEL\_SIZE, WHEEL\_SIZE, offset, 30);

}

}

}

* Output:



1. **write a Java Program that Accepts String Datafrom User and then Provide options forChanging case into Any of the Following. (UPPERCASE, lowercase,Sentence case. tOGGLE cASE)**

import java.util.Scanner;

public class StringCaseChanger {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

String original = input.nextLine();

System.out.println("What do you want to change the case to?");

System.out.println("1. Uppercase");

System.out.println("2. Lowercase");

System.out.println("3. Sentence case");

System.out.println("4. Toggle case");

System.out.print("Enter your choice: ");

int choice = input.nextInt();

switch (choice) {

case 1:

System.out.println("Uppercase: " + original.toUpperCase());

break;

case 2:

System.out.println("Lowercase: " + original.toLowerCase());

break;

case 3:

System.out.println("Sentence case: " + toSentenceCase(original));

break;

case 4:

System.out.println("Toggle case: " + toggleCase(original));

break;

default:

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

}

}

public static String toSentenceCase(String str) {

StringBuilder result = new StringBuilder();

boolean capitalizeNext = true;

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

char currentChar = str.charAt(i);

if (currentChar == ' ' || currentChar == '.') {

capitalizeNext = true;

} else if (capitalizeNext) {

result.append(Character.toUpperCase(currentChar));

capitalizeNext = false;

} else {

result.append(Character.toLowerCase(currentChar));

}

}

return result.toString();

}

public static String toggleCase(String str) {

StringBuilder result = new StringBuilder();

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

char currentChar = str.charAt(i);

if (Character.isUpperCase(currentChar)) {

result.append(Character.toLowerCase(currentChar));

} else if (Character.isLowerCase(currentChar)) {

result.append(Character.toUpperCase(currentChar));

} else {

result.append(currentChar);

}

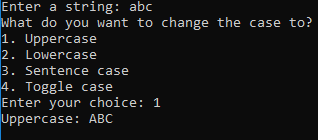
}

return result.toString();

}

}

* Output:



1. **Write an applet that simulates a Digital Clock displaying in the form Hour:Min: Sec. User should be able to change colour of Fonts of Clock by selection provided.**

/\*<applet code="digitalClock" width=200 height=60></applet>\*/

importjava.applet.\*;

importjava.awt.\*;

importjava.util.\*;

importjava.text.\*;

public class digitalClock extends Applet implements Runnable {

Thread t = null;

int hours=0, minutes=0, seconds=0;

String timeString = "";

public void init() {

setBackground(Color.green);

}

public void start() {

t = new Thread( this );

t.start();

}

public void run() {

try {

while (true) {

Calendar cal = Calendar.getInstance();

hours = cal.get( Calendar.HOUR\_OF\_DAY );

if ( hours > 12 ) hours -= 12;

minutes = cal.get( Calendar.MINUTE );

seconds = cal.get( Calendar.SECOND );

SimpleDateFormat formatter = new SimpleDateFormat("hh:mm:ss");

Date date = cal.getTime();

timeString = formatter.format( date );

repaint();

t.sleep( 1000 ); // interval given in milliseconds

}

}

catch (Exception e) { }

}

public void paint( Graphics g ) {

g.setColor(Color.blue );

g.drawString(timeString, 50, 50 );

}

}

* Output:



1. **Write a program to draw face Smiley with color using applet.**

importjava.awt.\*;

importjava.applet.\*;

/\*<applet code="cartoon" width="300" height="100"></applet>\*/

public class cartoon extends Applet

{

public void paint(Graphics g)

{

g.drawOval(40,40,150,150);//head

g.setColor(Color.yellow);

g.fillOval(40,40,150,150);

g.drawOval(62,75,30,30); //eye

g.drawOval(140,75,30,30); //eye

g.setColor(Color.red);

g.fillOval(57,75,30,30);

g.fillOval(138,75,30,30);

g.setColor(Color.white); //eye 2

g.fillOval(63,90,15,15);

g.fillOval(142,90,15,15);

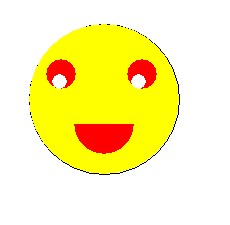
g.setColor(Color.red); //teeth

g.fillArc(85,110,60,60,180,180);

}

}

* Output:



1. **Write a program that accept Book information like Title, Author, Publication and Price for the N book from the user and display books in descending order with interval of 1 second using thread.**

import java.util.\*;

class Book extends Thread {

    String[] title;

    String[] author;

    String[] publication;

    Double[] price;

    public void getData(int num) {

        title = new String[num];

        author = new String[num];

        publication = new String[num];

        price = new Double[num];

        Scanner sc = new Scanner(System.in);

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

            System.out.print("Enter title of book: ");

            title[i] = sc.next();

            System.out.print("Enter name of author: ");

            author[i] = sc.next();

            System.out.print("Enter publication of book: ");

            publication[i] = sc.next();

            System.out.print("Enter price of book: ");

            price[i] = sc.nextDouble();

        }

    }

    public void printData(int num) {

        try {

            for (int i = num - 1; i >= 0; i--) {

                System.out.println("Title: " + title[i] + "\tAuthor: " + author[i] + "\tPublication: " + publication[i]

                        + "\tPrice: " + price[i]);

                Thread.sleep(1000);

            }

        } catch (Exception e) {

            System.out.println(e);

        }

    }

}

public class Assignment\_8 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

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

        int num = sc.nextInt();

        Book b1 = new Book();

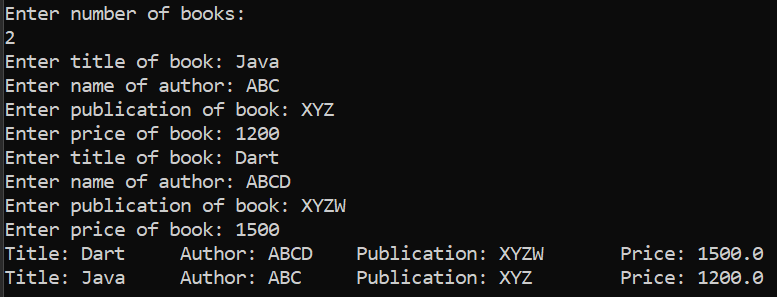
        b1.getData(num);

        b1.printData(num);

    }

}

* Output:



1. **Writer a program to demonstrate general operations of circular link list using switch case.**

import java.util.\*;

class Node {

    int data;

    Node next;

    public Node(int data) {

        this.data = data;

        this.next = null;

    }

}

public class Assignment\_9 {

    static Node last;

    public static void addToEmpty(int data) {

        if (last != null) {

            System.out.println("List is not empty");

            return;

        }

        Node temp = new Node(data);

        last = temp;

        last.next = last;

    }

    public static void addBegin(int data) {

        if (last == null) {

            addToEmpty(data);

            return;

        }

        Node temp = new Node(data);

        temp.next = last.next;

        last.next = temp;

    }

    public static void addEnd(int data) {

        if (last == null) {

            addToEmpty(data);

            return;

        }

        Node temp = new Node(data);

        temp.next = last.next;

        last.next = temp;

        last = temp;

    }

    public static void traverse() {

        if (last == null) {

            System.out.println("List is empty");

            return;

        }

        Node temp = last.next;

        while (temp != last) {

            System.out.print(temp.data + " ");

            temp = temp.next;

        }

        System.out.println(temp.data);

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        while (true) {

            System.out.println("1. Add to empty list");

            System.out.println("2. Add at beginning");

            System.out.println("3. Add at end");

            System.out.println("4. Traverse");

            System.out.println("5. Quit");

            System.out.print("Enter your choice : ");

            int choice = sc.nextInt();

            switch (choice) {

                case 1:

                    System.out.print("Enter integer element to insert : ");

                    int data = sc.nextInt();

                    addToEmpty(data);

                    break;

                case 2:

                    System.out.print("Enter integer element to insert : ");

                    data = sc.nextInt();

                    addBegin(data);

                    break;

                case 3:

                    System.out.print("Enter integer element to insert : ");

                    data = sc.nextInt();

                    addEnd(data);

                    break;

                case 4:

                    traverse();

                    break;

                case 5:

                    System.exit(0);

                default:

                    System.out.println("Wrong choice");

            }

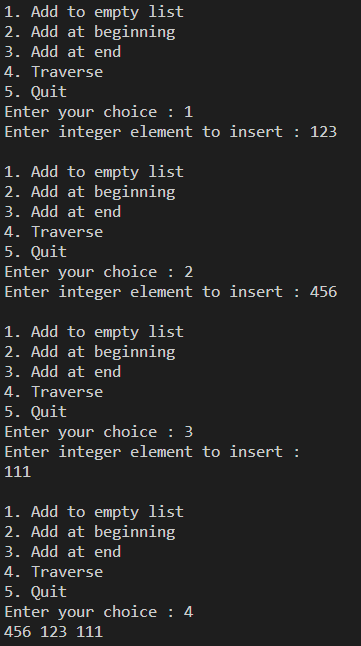
            System.out.println();

        }

    }

}

* Output:



1. **Writer a program to demonstrate general operations of singly link list using switch case.**

class Node {

    int value;

    Node next;

    public Node(int value) {

        this.value = value;

        this.next = null;

    }

}

public class Assignment\_10 {

    Node start;

    public void insert(int value) {

        Node current = start;

        while (current.next != null) {

            current = current.next;

        }

        Node newNode = new Node(value);

        current.next = newNode;

    }

    public void printvalues() {

        Node current = start;

        while (current.next != null) {

            System.out.println(current.value);

            current = current.next;

        }

        System.out.println(current.value);

    }

    public void deleteNode(int value) {

        if (value == start.value) {

            start = start.next;

            return;

        }

        Node current = start;

        while (current.next.value != value) {

            current = current.next;

        }

        System.out.println("Node Deleted" + value);

        current.next = current.next.next;

    }

    public void searchNode(int value) {

        Node current = start;

        int position = 0;

        while (current != null) {

            if (current.value == value)

                System.out.println("position of search node" + current.value + " is => " + position);

            current = current.next;

            position++;

        }

    }

    public static void main(String args[]) {

        Assignment\_10  singlyLinkedList = new Assignment\_10();

        singlyLinkedList.start = new Node(5);

        singlyLinkedList.insert(6);

        singlyLinkedList.insert(7);

        singlyLinkedList.insert(8);

        singlyLinkedList.insert(9);

        singlyLinkedList.insert(10);

        singlyLinkedList.insert(11);

        singlyLinkedList.printvalues();

        singlyLinkedList.deleteNode(10);

        System.out.println("after delete list is=>");

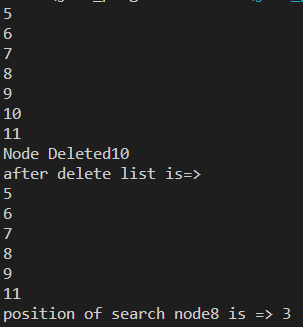
        singlyLinkedList.printvalues();

        singlyLinkedList.searchNode(8);

    }

}

* Output:



1. **Write a program which demonstrate sunset using an Applet.**

import java.applet.Applet; import java.awt.\*; import java.util.Timer; import java.util.TimerTask;

/\* <applet code="program11" height=500 width=700> </applet> \*/ public class program11 extends Applet { private int sunRadius = 50; private int sunX = 100; private int sunY = 100; private int sunSpeed = 1; private int skyY = 0; private int skySpeed = 1;

public void init() { setBackground(Color.CYAN); Timer timer = new Timer(); timer.schedule(new TimerTask() {

public void run() { sunY += sunSpeed; skyY += skySpeed; if (skyY >= getSize().height) {

skyY = getSize().height;

} repaint();

}

}, 0, 50);

}

public void paint(Graphics g) {

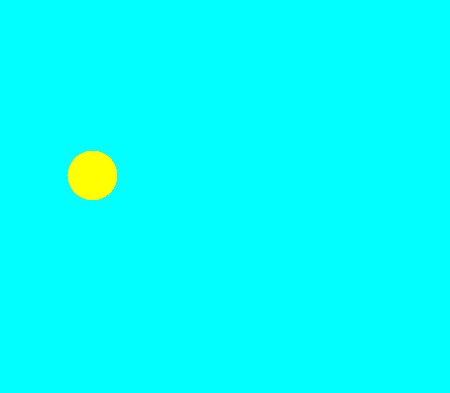
g.setColor(Color.YELLOW);

g.fillOval(sunX, sunY, sunRadius, sunRadius);

}

}

Output:



1. **Write a program add/sub/mul/div of 2 numbers in Applet using button.**

import java.applet.Applet; import java.awt.\*; import java.awt.event.\*;

/\* <applet code="program12" height=500 width=700> </applet> \*/ public class program12 extends Applet implements ActionListener { private TextField num1Field, num2Field, resultField;

private Button addButton, subtractButton, multiplyButton, divideButton; public void init() { num1Field = new TextField(10); num2Field = new TextField(10); resultField = new TextField(10); resultField.setEditable(false); addButton = new Button("+"); addButton.addActionListener(this); subtractButton = new Button("-"); subtractButton.addActionListener(this); multiplyButton = new Button("x"); multiplyButton.addActionListener(this); divideButton = new Button("/"); divideButton.addActionListener(this);

add(num1Field); add(num2Field); add(addButton); add(subtractButton); add(multiplyButton); add(divideButton); add(resultField);

}

public void actionPerformed(ActionEvent e) { double num1 = Double.parseDouble(num1Field.getText()); double num2 = Double.parseDouble(num2Field.getText()); double result = 0;

if (e.getSource() == addButton) { result = num1 + num2;

} else if (e.getSource() == subtractButton) { result = num1 - num2;

} else if (e.getSource() == multiplyButton) { result = num1 \* num2;

} else if (e.getSource() == divideButton) { result = num1 / num2;

}

resultField.setText(Double.toString(result));

}

}

Output:



1. Accept N number from applet tag and print that many numbers of object using an Applet.

import java.applet.Applet; import java.awt.\*;

/\*<applet code="program13" width="500" height="500">

<param name="numberOfObjects" value="10">

</applet>\*/ public class program13 extends Applet { private int numberOfObjects = 0;

public void init() {

String numberOfObjectsStr = getParameter("numberOfObjects"); if (numberOfObjectsStr != null) {

numberOfObjects = Integer.parseInt(numberOfObjectsStr);

}

}

public void paint(Graphics g) { for (int i = 0; i<numberOfObjects; i++) {

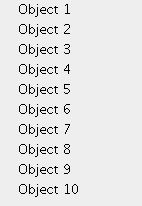
g.drawString("Object " + (i + 1), 20, 20 \* (i + 1));

}

}

}

Output:



1. **Write a program which demonstrate functionality of creation and use of package.**

// File: com/example/util/MathUtil.java

package com.example.util;

public class MathUtil {

public static int add(int a, int b) {

return a + b;

}

public static int subtract(int a, int b) {

return a - b;

}

}

// File: com/example/Main.java

package com.example;

import com.example.util.MathUtil;

public class Main 007B

public static void main(String[] args) {

int a = 10;

int b = 20;

System.out.println("The sum of " + a + " and " + b + " is: " + MathUtil.add(a, b));

System.out.println("The difference of " + a + " and " + b + " is: " + MathUtil.subtract(a, b));

} }



1. **Accept number from applet tag parameter and draw that many numbers of objects. Fill them with different color.**

/\*<applet code="DrawingApplet.class" width="500" height="500">

<param name="numberOfShapes" value0">

</applet>\*/

import java.applet.Applet;

import java.awt.\*;

public class DrawingApplet extends Applet {

private int numberOfShapes = 0;

public void init() {

String numberOfShapesStr = getParameter("numberOfShapes");

if (numberOfShapesStr != null) {

numberOfShapes = Integer.parseInt(numberOfShapesStr);

}

}

public void paint(Graphics g) {

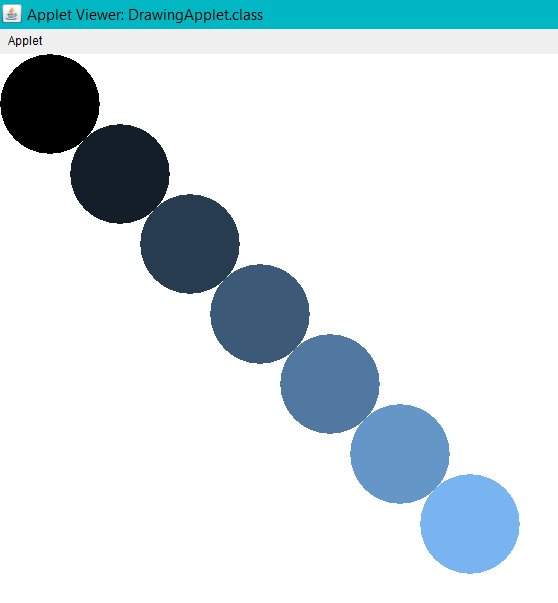
for (int i = 0; i < numberOfShapes; i++) {

g.setColor(new Color(i \* 20, i \* 30, i \* 40));

g.fillOval(i \* 70, i \* 70, 100, 100);

}

} }

****



importjava.applet.\*;

public class circleandtrangle extends Applet

{

int x1[]={405,262,300,489,543};

int y1[]={149,245,413,422,252};

int n1=5;

int x2[]={813,661,710,892,950};

int y2[]={132,250,418,425,258};

int n2=5;

public void paint (Graphics g)

{

g.setColor(Color.red);

g.fillArc(250,150,300,300,0,360);

g.setColor(Color.black);

g.fillPolygon(x1,y1,n1);

g.setColor(Color.white);

g.drawString("Applet",380,300);//

g.setColor(Color.red);

g.fillPolygon(x2,y2,n2);

g.setColor(Color.black);

g.fillArc(680,170,250,250,0,360);

g.setColor(Color.white);

g.drawString("hello",790,300);} }

