**Name:**

**DIV: Roll No:**

**Subject: Java Practical Assignment (403)**

**Index**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Description** | **Page No.** | **Date Of**  **Submission** | **Sig n** |
| 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 | 3 |  |  |
| 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 | 5 |  |  |
| 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. | 6 |  |  |
| 4 | Write an applet that simulates a Rotating wheel. User can start and stop rotation by clicking Start and Stop Button. | 8 |  |  |
| 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). | 10 |  |  |
| 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. | 12 |  |  |
| 7 | Write a program to draw Smiley face with color using applet | 14 |  |  |
| 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. | 15 |  |  |
| 9 | Writer a program to demonstrate general operations of circular link list using switch case. | 17 |  |  |
| 10 | Writer a program to demonstrate general operations of singly link list using switch case. | 20 |  |  |
| 11 | Write a program which demonstrate sunset using an Applet. | 23 |  |  |
| 12 | Write a program add/sub/mul/div of 2 numbers in |  |  |  |
|  | Applet using button. | 25 |  |  |
| 13 | Accept N number from applet tag and print that many numbers of object using an Applet. | 27 |  |  |
| 14 | Write a program which demonstrate functionality of creation and use of package. | 28 |  |  |
| 15 | Accept number from applet tag parameter and draw that many numbers of objects. Fill them with different color. | 30 |  |  |
|  |  |  |  |  |

**Date:**

**Assign mentor’s Signature:**

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.**

Code:

import java.util.\*; class Student extends Thread {

int RollNo;

String name;

public void getData() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Roll No of Student: ");

RollNo = sc.nextInt();

System.out.println("Enter name of Student: "); name = sc.next();

}

public void printData() {

try {

System.out.println("Roll: " + RollNo + " \tName: " + name);

Thread.sleep(1000);

} catch (Exception e) {

System.out.println(e);

}

}

}

public class Program1 {

public static void main(String args[]) { Student s1 = new Student(); s1.getData();

Student s2 = new Student(); s2.getData();

Student s3 = new Student(); s3.getData();

Student s4 = new Student(); s4.getData();

Student s5 = new Student();

s5.getData();

System.out.println("Students: "); s1.printData();

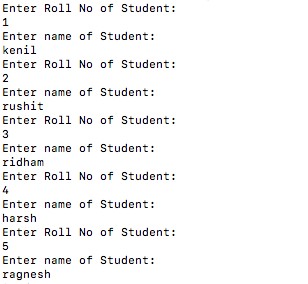
s2.printData(); s3.printData(); s4.printData();

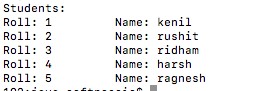
s5.printData();

}

}

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.**

Code:

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);

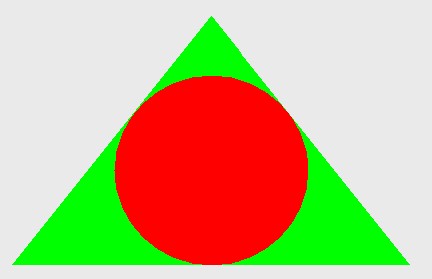
g.setColor(Color.red);

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.**

Code:

import java.io.\*; class Letters {

public void printVovels(String str, int choise) { 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 (choise == 1) {

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

} else {

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

}

}

}

}

}

public void printNonVovels(String str, int choise) { 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 (choise == 1) {

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

} else {

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

}

}

}

}

}

}

public class Program3 { 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 choise: ");

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

System.out.println("For Non vovels: 2"); String a = br.readLine(); int choise1 = Integer.parseInt(a);

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

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

System.out.println("For Lowercase: 2"); String b = br.readLine(); int choise2 = Integer.parseInt(b); if (choise1 == 1) {

l.printVovels(str, choise2);

} else if (choise1 == 2) {

l.printNonVovels(str, choise2);

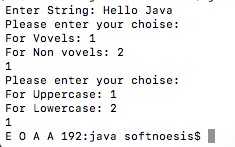
} else {

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

}

} }

Output:



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

Code:

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

// <applet code="program4" height=500 width=700> </applet> public class program4 extends Applet implements ActionListener, Runnable { private Button startButton; private Button stopButton; private Boolean stopFlag; private Thread animationThread;

private int angle; public void init() { startButton = new Button("Start"); stopButton = new Button("Stop");

add(startButton); add(stopButton); startButton.addActionListener(this); stopButton.addActionListener(this);

stopFlag = true;

animationThread = null;

}

public void start() { if (animationThread == null) {

stopFlag = false;

animationThread = new Thread(this);

animationThread.start();

}

}

public void stop() { stopFlag = true;

}

public void run() { while (!stopFlag) { angle += 5; repaint(); try { Thread.sleep(100);

} catch (InterruptedException e) {

}

}

}

public void actionPerformed(ActionEvent e) {

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

start();

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

}

}

public void paint(Graphics g) { int x = getSize().width / 2; int y = getSize().height / 2; g.setColor(Color.black);

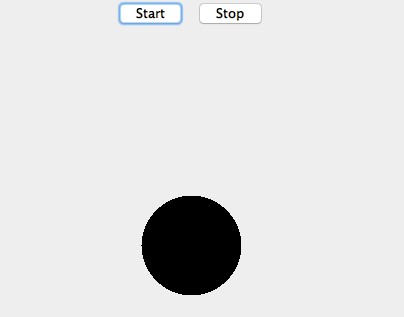
g.fillOval(x - 50, y - 50, 100, 100);

g.setColor(Color.red); int x2 = (int) (x + 40 \* Math.cos(Math.toRadians(angle))); int y2 = (int) (x - 40 \* Math.sin(Math.toRadians(angle))); g.fillRect(x2 - 10, y2 - 2, 20, 4);

}

};

Output:



1. **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).**

Code:

import java.io.\*; public class Program5 { 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();

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

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

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

System.out.println("For Sentencecase: 3");

System.out.println("For toggalcase: 4"); String c = br.readLine(); int choise = Integer.parseInt(c); switch (choise) { case 1:

System.out.println(str.toUpperCase()); break; case 2:

System.out.println(str.toLowerCase()); break; case 3:

for (int i = 0; i < str.length(); i++) { if (Character.isSpaceChar(str.charAt(i))) {

str = str.replace(str.charAt(i + 1), Character.toUpperCase(str.charAt(i + 1)));

}

str = str.replace(str.charAt(0), Character.toUpperCase(str.charAt(0)));

}

System.out.print(str);

break; case 4:

str = str.toUpperCase();

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

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

str = str.replace(str.charAt(i + 1), Character.toLowerCase(str.charAt(i + 1)));

}

str = str.replace(str.charAt(0), Character.toLowerCase(str.charAt(0)));

}

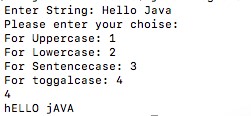
System.out.println(str);

break; default:

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

} }

Output:



1. **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.**

Code:

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

import javafx.scene.control.Button; import java.text.\*;

// <applet code="DigitalClock" width="200" height="200"> </applet public class DigitalClock extends Applet implements Runnable { Thread t = null;

int hours = 0, minutes = 0, seconds = 0, am = 1;

String timeString = ""; public void init() {

setBackground(Color.white);

}

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; am = 0;

}

minutes = cal.get(Calendar.MINUTE); seconds = cal.get(Calendar.SECOND);

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

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.BLACK);

g.drawRect(30, 30, 100, 50);

g.setColor(Color.red);

g.drawString(timeString, 40, 60);

g.setColor(Color.blue); Button button1 = new Button("Bule");

}

} Output:



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

Code:

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

/\*<applet code="program7" width=2000 height=2000></applet>\*/

public class program7 extends Applet

{

public void paint(Graphics g)

{

g.setColor(Color.yellow);

g.fillOval(400,150,200,200);

g.setColor(Color.black);

g.fillOval(430,200,40,40);

g.fillOval(530,200,40,40);

g.setColor(Color.white);

g.fillOval(440,220,20,20);

g.fillOval(540,220,20,20);

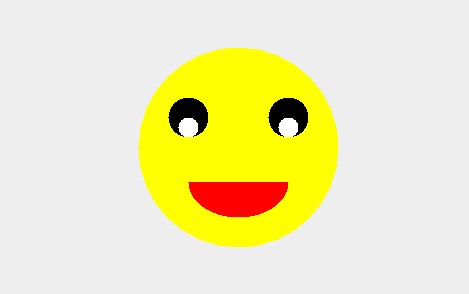
g.setColor(Color.red);

g.fillArc(450,250,100,70,180,180);

}

}

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.**

Code:

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.println("Enter title of book: "); title[i] = sc.next();

System.out.println("Enter name of author: "); author[i] = sc.next();

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

publication[i] = sc.next();

System.out.println("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 Program8 { public static void main(String args[]) { Scanner sc = new Scanner(System.in); System.out.println("Enter number of books: ");

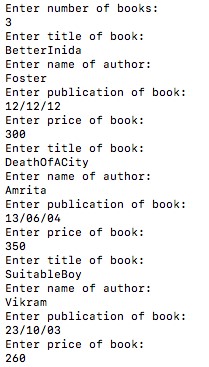
int num = sc.nextInt(); Book b1 = new Book(); b1.getData(num);

b1.printData(num);

}

}

Output:





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

Code:

import java.util.\*;

class Node {

int data;

Node next; public Node(int data) { this.data = data; this.next = null;

}

} public class program9 { 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");

}

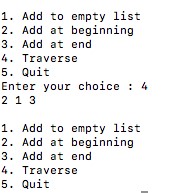
System.out.println();

}

}

}

Output:



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

Code:

class Node

{ int value; Node next;

public Node(int value)

{

this.value=value; this.next=null;

}

}

public class singlyLinkedList1

{

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[]){

singlyLinkedList1 singlyLinkedList = new singlyLinkedList1(); singlyLinkedList.start=new Node(5);

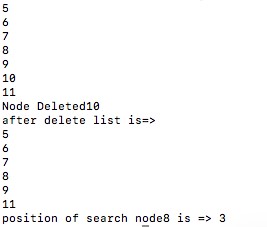
singlyLinkedList.insert(6); singlyLinkedList.insert(7); singlyLinkedList.insert(8); singlyLinkedList.insert(9); singlyLinkedList.insert(10); singlyLinkedList.insert(11); singlyLinkedList.printvalues(); singlyLinkedList.deleteNode(10);

System.out.println("after delete list is=>"); singlyLinkedList.printvalues(); singlyLinkedList.searchNode(8);

}

}

Output:



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

Code:

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) {

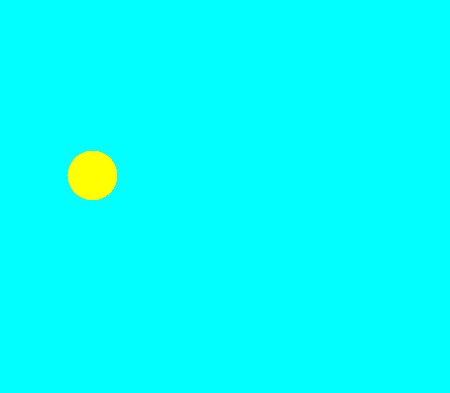
g.setColor(Color.YELLOW);

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

}

}

Output:



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

Code:

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.

Code:

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++) {

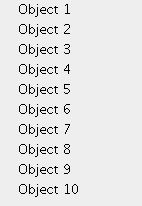
g.drawString("Object " + (i + 1), 20, 20 \* (i + 1));

}

}

}

Output:



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

Code:

package calculator; public class Calc {

public void Sum(int a, int b) {

System.out.println("Sum of two number is: " + (a + b));

}

public void Sub(int a, int b) {

System.out.println("Subtraction of two number is: " + (a - b));

}

public void Mul(int a, int b) {

System.out.println("Multiplication of two number is: " + (a \* b));

}

public void Div(int a, int b) {

System.out.println("Divition of two number is: " + (a / b));

}

}

// CalculatorDemo.java

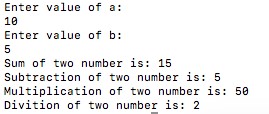
import calculator.Calc; import java.util.\*; public class CalculatorDemo { public static void main(String args[]) { Scanner sc = new Scanner(System.in); System.out.println("Enter value of a: "); int a = sc.nextInt();

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

int b = sc.nextInt(); Calc clac = new Calc(); clac.Sum(a, b); clac.Sub(a, b); clac.Mul(a, b); clac.Div(a, b);

}

} Output:



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

Code:

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

/\*<applet code="program15" width=2000 height=2000></applet>\*/ public class program15 extends Applet

{ int x1[]={115,215,300,215,115}; int y1[]={150,100,200,297,250}; int n1=5;

int x2[]={500,600,550,450,405}; int y2[]={70,150,250,250,150}; int n2=5;

public void paint(Graphics g)

{

g.setColor(Color.red);

g.fillOval(100,100,200,200);

g.setColor(Color.blue);

g.fillPolygon(x1,y1,n1);

g.fillPolygon(x2,y2,n2);

g.setColor(Color.black);

g.drawString("Applet",180,200);

g.setColor(Color.red);

g.fillOval(425,93,155,155);

g.setColor(Color.black);

g.drawString("Hello",485,180);

}

}

Output:

