==

SUTEX BANK COLLEGE OF COMPUTER APPLICATIONS & SCIENCE,

AMROLI

S.Y.B.C.A. PROGRAMME-2022-23

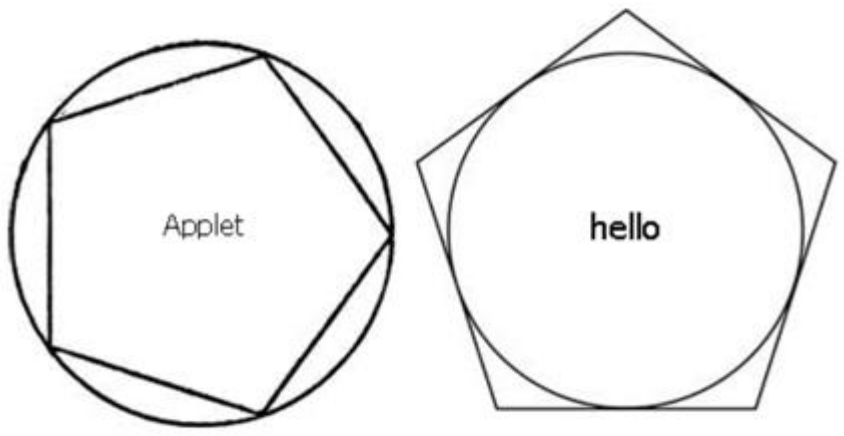
PRACTICAL ASSIGNMENT

SUBJECT: 403: JAVA PROGRAMMING LANGUAGE

SEMESTER:IV

==

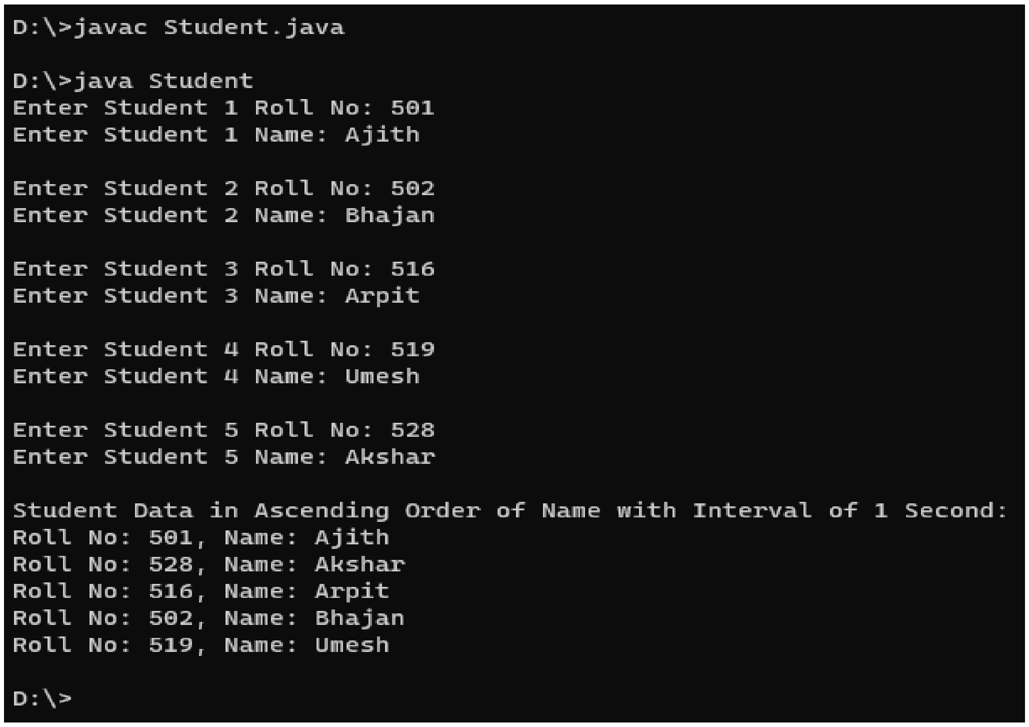
Sr No	Programs	Sign
1	create student class having data members roll_no and name.create 5 objects of student class and take input from the user and print all student data in ascending order of name with interval of 1 second in java .	
2	Create an Applet which displays a Triangle Within a circle where the circumference of the circle touches to the all vertices of the triangle.Provide different colors to both the objects. in java.	
3	write a java program that accepts string data . extract either all vowels or consonants 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 Strings 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. user should be able to change color of fonts of clock by selection provided .	
7	write a program to draw face Smiley with color using applet in java.	

8	write a program that accept book information like title,author,publication and price for the N book from the user and display book 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.	
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.	
		

Answers :-

```
1. import java.util.Scanner;
   public class Student {
        String roll_no;
        String name;
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                Student[] stud = new Student[5];
                for(int i=0; i<5; i++) {
                        stud[i] = new Student();
                        System.out.print("Enter Student " + (i+1) + " Roll No: ");
                        stud[i].roll_no = sc.next();
                        System.out.print("Enter Student " + (i+1) + " Name: ");
                        stud[i].name = sc.next();
                        System.out.println();
                }
                for(int i=0; i<5; i++) {
                        for(int j=0; j<4; j++) {
                                if(stud[j].name.compareTo(stud[j+1].name) > 0) {
                                        Student temp = stud[j];
                                        stud[j] = stud[j+1];
                                        stud[j+1] = temp;
                                }
                        }
                }
                System.out.println("Student Data in Ascending Order of Name with
Interval of 1 Second:");
                for(int i=0; i<5; i++) {
                        try {
                                Thread.sleep(1000);
                                System.out.println("Roll No: " + stud[i].roll_no + ", Name: " +
stud[i].name);
                        } catch (InterruptedException e) {
                                e.printStackTrace();
                        }
                }
        }
}
```

Output :



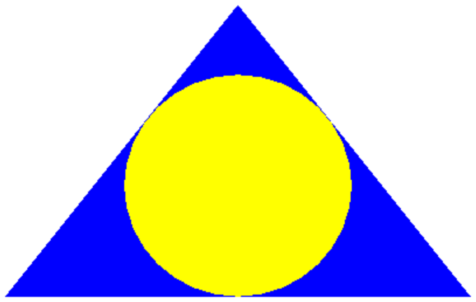
```
D:\>javac Student.java

D:\>java Student
Enter Student 1 Roll No: 501
Enter Student 1 Name: Ajith

Enter Student 2 Roll No: 502
Enter Student 2 Name: Bhajan

Enter Student 3 Roll No: 516
Enter Student 3 Name: Arpit

Enter Student 4 Roll No: 519
Enter Student 4 Name: Umesh

Enter Student 5 Roll No: 528
Enter Student 5 Name: Akshar

Student Data in Ascending Order of Name with Interval of 1 Second:
Roll No: 501, Name: Ajith
Roll No: 528, Name: Akshar
Roll No: 516, Name: Arpit
Roll No: 502, Name: Bhajan
Roll No: 519, Name: Umesh

D:\>
```

==

2.

```
/* <applet code="que2" height=500 width=700></applet> */
import java.awt.*;
import java.applet.*;

public class que2 extends Applet
{
  public void paint(Graphics g)
  {
        //Triangle
        g.setColor(Color.blue);
        int x[] = {350, 150, 550};
        int y[] = {100, 350, 350};
        int n = 3;
        g.fillPolygon(x, y, n);

        //oval
        g.setColor(Color.yellow);
        g.fillOval(253, 160, 195, 190); } };
```

Output:

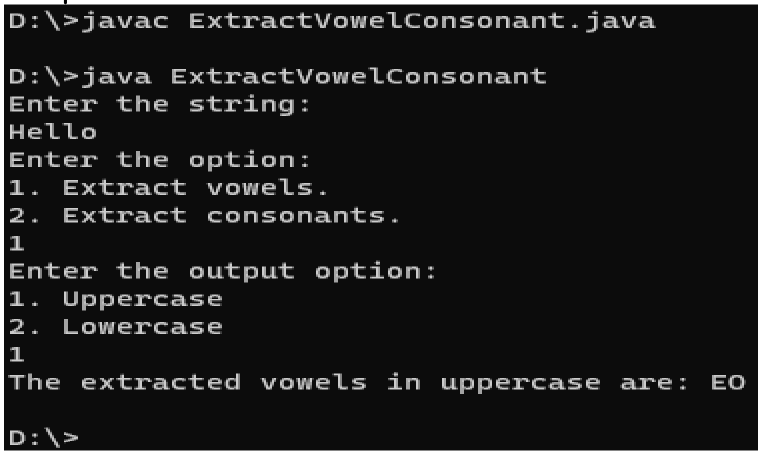


==
3.

```
import java.util.Scanner;
public class ExtractVowelConsonant {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string:");
        String str = sc.nextLine();
        System.out.println("Enter the option:");
        System.out.println("1. Extract vowels.");
        System.out.println("2. Extract consonants.");
        int option = sc.nextInt();
        switch(option){
            case 1:
                extractVowels(str);
                break;
            case 2:
                extractConsonants(str);
                break;
            default:
                System.out.println("Invalid option!");
        }
    }

    public static void extractVowels(String str){
        System.out.println("Enter the output option:");
        System.out.println("1. Uppercase");
        System.out.println("2. Lowercase");
        Scanner sc = new Scanner(System.in);
        int option = sc.nextInt();
        String vowels = "";
        switch(option){
            case 1:
                for(int i=0;i<str.length();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'){
                        vowels = vowels + str.charAt(i);
                    }
                }
                System.out.println("The extracted vowels in uppercase are:
"+vowels.toUpperCase());
                break;
            case 2:
                for(int i=0;i<str.length();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'){
                        vowels = vowels + str.charAt(i);
                    }
                }
                System.out.println("The extracted vowels in lowercase are:
"+vowels.toLowerCase());
                break;
            default:
                System.out.println("Invalid option!");
        }
    }

    public static void extractConsonants(String str){
        System.out.println("Enter the output option:");
        System.out.println("1. Uppercase");
        System.out.println("2. Lowercase");
        Scanner sc = new Scanner(System.in);
        int option = sc.nextInt();
        String consonants = "";
        switch(option){
            case 1:
```

```
            for(int i=0;i<str.length();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' && str.charAt(i)!=' '){
                  consonants = consonants + str.charAt(i);
               }
            }
            System.out.println("The extracted consonants in uppercase are:
"+consonants.toUpperCase());
            break;
         case 2:
            for(int i=0;i<str.length();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' && str.charAt(i)!=' '){
                  consonants = consonants + str.charAt(i);
               }
            }
            System.out.println("The extracted consonants in lowercase are:
"+consonants.toLowerCase());
            break;
         default:
            System.out.println("Invalid option!");
      }
   }
}
```

Output :



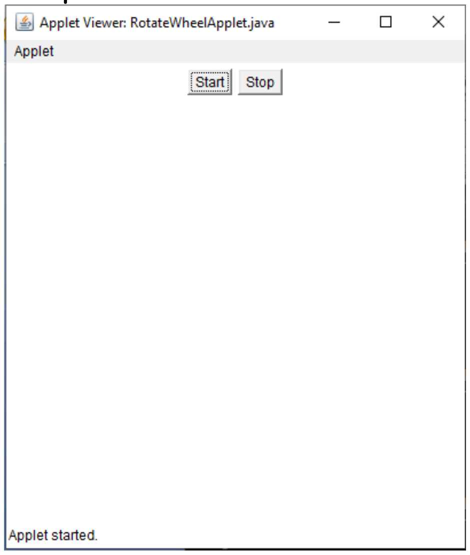
```
D:\>javac ExtractVowelConsonant.java

D:\>java ExtractVowelConsonant
Enter the string:
Hello
Enter the option:
1. Extract vowels.
2. Extract consonants.
1
Enter the output option:
1. Uppercase
2. Lowercase
1
The extracted vowels in uppercase are: EO

D:\>
```

==

4.
```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Button;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
/*
<applet code = "RotateWheelApplet.java" height=400 width=400>
</applet>
*/
public class RotateWheelApplet extends Applet implements ActionListener {
   private int rotateAngle;
   private Button startButton;
   private Button stopButton;
   // Initialize the applet
   public void init() {
      rotateAngle = 0;
      startButton = new Button("Start");
      stopButton = new Button("Stop");
      add(startButton);
      add(stopButton);
      // Register the action listener
      startButton.addActionListener(this);
      stopButton.addActionListener(this);
   }
   // Paint the applet
   public void paint(Graphics g) {
      g.drawArc(50, 50, 200, 200, 0, rotateAngle);
   }
   // Handle the action event
   public void actionPerformed(ActionEvent event) {
      if (event.getSource() == startButton) {
         for (int i = 0; i < 360; i++) {
            rotateAngle = (rotateAngle + 1) % 360;
            repaint();
            try {
               Thread.sleep(20);
            } catch(InterruptedException e) {
```

```
            e.printStackTrace();
          }
        }
      }
      else if (event.getSource() == stopButton) {
        rotateAngle = 0;
        repaint();
      }
    }
}
```

Output :



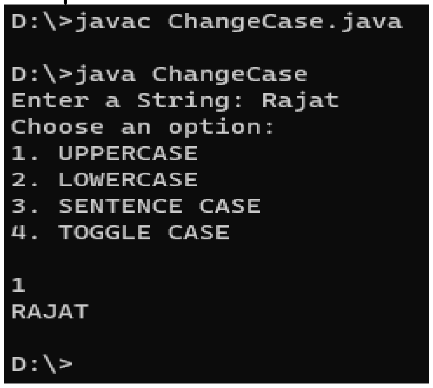
Applet started.

==

5.

```
import java.util.Scanner;
public class ChangeCase {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a String: ");
        String inputString = scanner.nextLine();

        System.out.println("Choose an option:\n1. UPPERCASE\n2. LOWERCASE\n3.
SENTENCE CASE\n4. TOGGLE CASE\n");
        int option = scanner.nextInt();
        scanner.close();
        switch (option) {
            case 1:
                System.out.println(inputString.toUpperCase());
                break;
            case 2:
                System.out.println(inputString.toLowerCase());
                break;
            case 3:
                System.out.println(inputString.substring(0, 1).toUpperCase() +
inputString.substring(1).toLowerCase());
                break;
            case 4:
                StringBuilder result = new StringBuilder();
                for (int i = 0; i < inputString.length(); i++) {
                    char currentChar = inputString.charAt(i);
                    if (Character.isUpperCase(currentChar)) {
                        result.append(Character.toLowerCase(currentChar));
                    } else {
                        result.append(Character.toUpperCase(currentChar));
                    }
                }
                System.out.println(result);
                break;
            default:
                System.out.println("Invalid Option!");
                break;
        }
    }
}
```

Output :

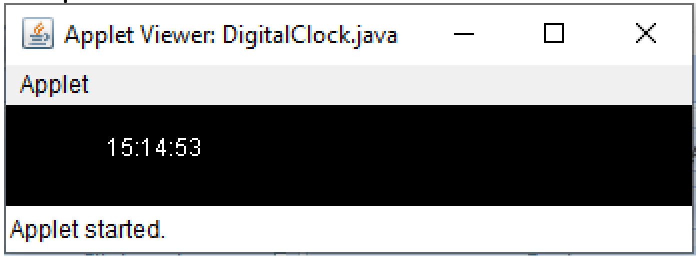


```
D:\>javac ChangeCase.java

D:\>java ChangeCase
Enter a String: Rajat
Choose an option:
1. UPPERCASE
2. LOWERCASE
3. SENTENCE CASE
4. TOGGLE CASE

1
RAJAT

D:\>
```

===
6.
```
import java.awt.*;
import java.applet.*;
import java.util.*;

/*
<applet code="DigitalClock.java" height=400 width=400>
</applet>
*/

public class DigitalClock extends Applet implements Runnable {
  String timeString = "";
  Thread clockThread;
  Color fontColor;

  public void init() {
    setBackground(Color.black);
    setSize(200, 50);
    fontColor = Color.white;
  }

  public void start() {
    clockThread = new Thread(this);
    clockThread.start();
  }

  public void stop() {
    clockThread = null;
  }

  public void run() {
    Thread thisThread = Thread.currentThread();
    while (clockThread == thisThread) {
      try {
        Calendar cal = Calendar.getInstance();
        int hour = cal.get(Calendar.HOUR_OF_DAY);
        int min = cal.get(Calendar.MINUTE);
```

```
            int sec = cal.get(Calendar.SECOND);
            timeString = hour + ":" + min + ":" + sec;
            repaint();
            Thread.sleep(1000);
          } catch (InterruptedException e) {
            // do nothing
          }
        }
      }

      public void paint(Graphics g) {
        g.setColor(fontColor);
        g.drawString(timeString, 50, 25);
      }

      //User can change the color of the font
      public void setFontColor(Color c) {
        fontColor = c;
      }

}
```

Output :



```
15:14:53
```

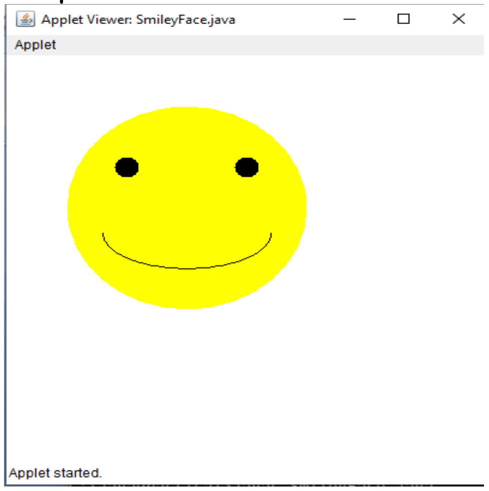
Applet started.

==

7.

```
/*
<applet code="SmileyFace.java" height=400 width=400></applet>
*/
import java.applet.Applet;
import java.awt.*;
public class SmileyFace extends Applet
{
   public void paint (Graphics g)
   {
      g.setColor (Color.yellow); // set color to yellow
      g.fillOval (50, 50, 200, 200); // draw face
      g.setColor (Color.black); // set color to black
      g.fillOval (90, 100, 20, 20); // draw left eye
      g.fillOval (190, 100, 20, 20); // draw right eye
      g.drawArc (80, 140, 140, 70, 0, -180); // draw smile
   }
}
```

Output :



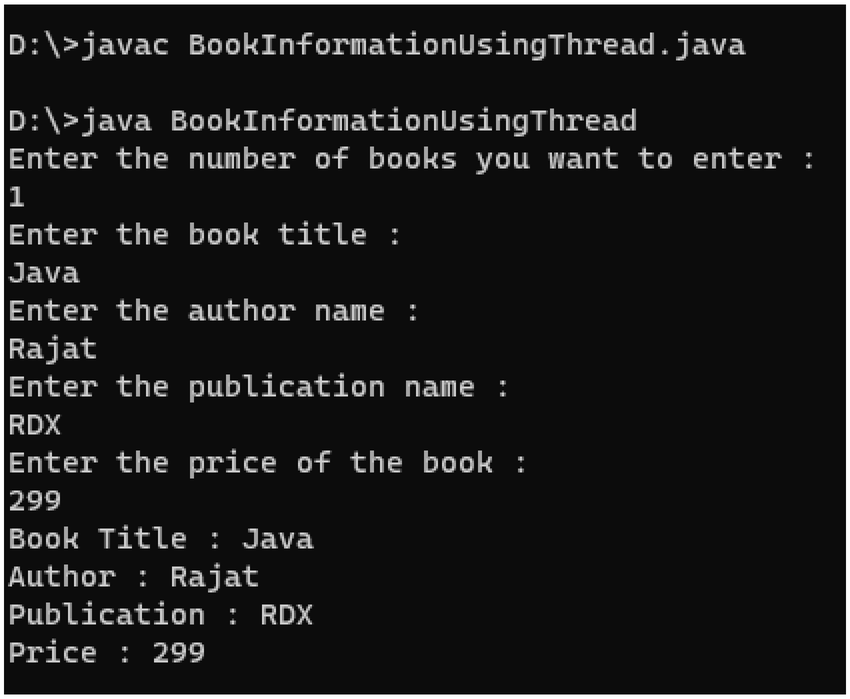
```
Applet started.
```

==

8.

```
import java.util.Scanner;

public class BookInformationUsingThread {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number of books you want to enter : ");
                int N = sc.nextInt();
                String[][] bookDetails = new String[N][4];
                for(int i=0; i<N; i++) {
                        System.out.println("Enter the book title : ");
                        bookDetails[i][0] = sc.next();
                        System.out.println("Enter the author name : ");
                        bookDetails[i][1] = sc.next();
                        System.out.println("Enter the publication name : ");
                        bookDetails[i][2] = sc.next();
                        System.out.println("Enter the price of the book : ");
                        bookDetails[i][3] = sc.next();
                }
                Thread sortThread = new Thread() {
        @Override
        public void run() {
            sortBookDetails(bookDetails, N);
        }
    };
    sortThread.start();
      }
      public static void sortBookDetails(String[][] bookDetails, int N) {
                for(int i=0; i<N; i++) {
                        for(int j=0; j<N-1; j++) {
                                int price1 = Integer.parseInt(bookDetails[j][3]);
                                int price2 = Integer.parseInt(bookDetails[j+1][3]);
                                if(price1<price2) {
                                        String[] temp = bookDetails[j];
                                        bookDetails[j] = bookDetails[j+1];
                                        bookDetails[j+1] = temp;
                                }
                        }
                        System.out.println("Book Title : "+bookDetails[i][0]+"\nAuthor :
"+bookDetails[i][1]+"\nPublication : "+bookDetails[i][2]+"\nPrice : "+bookDetails[i][3]);
                        try {
                                Thread.sleep(1000);
                        } catch (InterruptedException e) {
                                e.printStackTrace();
                        }
                }
        }
}
```

Output :



```
D:\>javac BookInformationUsingThread.java

D:\>java BookInformationUsingThread
Enter the number of books you want to enter :
1
Enter the book title :
Java
Enter the author name :
Rajat
Enter the publication name :
RDX
Enter the price of the book :
299
Book Title : Java
Author : Rajat
Publication : RDX
Price : 299
```

==

9.

```
import java.util.*;
public class CircularLinkedList {

   static class Node {
      int data;
      Node next;
   }

   static Node head = null;
   static Node tail = null;
   static int size = 0;

   // add element at the end of the list
   static void add(int data)
   {
      Node node = new Node();
      node.data = data;
      node.next = null;

      if (head == null) {
         head = node;
```

```
            tail = node;
            node.next = head;
        }
        else {
            tail.next = node;
            tail = node;
            tail.next = head;
        }
        size++;
    }
    // insert element at the start of the list
    static void insertAtStart(int data)
    {
        Node node = new Node();
        node.data = data;
        node.next = head;
        head = node;
        tail.next = node;
        size++;
    }
    // insert element at the given index
    static void insertAt(int index, int data)
    {
        if (index == 0) {
            insertAtStart(data);
            return;
        }
        if (index == size) {
            add(data);
            return;
        }
        Node node = new Node();
        node.data = data;
        Node temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        node.next = temp.next;
        temp.next = node;
        size++;
    }

    // delete element at the given index
    static void deleteAt(int index)
    {
        if (index == 0) {
            head = head.next;
            tail.next = head;
            size--;
            return;
        }
        if (index == size - 1) {
            Node s = head;
            Node t = head;
            while (s != tail) {
                t = s;
                s = s.next;
            }
            tail = t;
            tail.next = head;
            size--;
            return;
        }
        Node temp1 = head;
        index = index - 1;
        for (int i = 0; i < index; i++) {
            temp1 = temp1.next;
        }
        temp1.next = temp1.next.next;
        size--;
    }

    // print the list
    static void printList()
    {
        Node temp = head;
        if (size == 0) {
            System.out.print("empty\n");
            return;
        }
        if (head.next == head) {
            System.out.println(head.data);
            return;
        }
        System.out.print(head.data + "->");
        temp = head.next;
        while (temp != head) {
            System.out.print(temp.data + "->");
            temp = temp.next;
        }
        System.out.println("...");
    }

    public static void main(String[] args)
    {
        int choice = 0;
        Scanner sc = new Scanner(System.in);
        while (choice != 6) {
            System.out.println("1. Add element at the end of the list");
            System.out.println("2. Insert element at the start of the list");
            System.out.println("3. Insert element at the given index");
            System.out.println("4. Delete element at the given index");
            System.out.println("5. Print List");
            System.out.println("6. Exit");
            System.out.println("Enter your choice : ");
            choice = sc.nextInt();
            switch (choice) {
            case 1:
                System.out.println("Enter element to add : ");
                int data = sc.nextInt();
                add(data);
                break;
            case 2:
                System.out.println("Enter element to insert : ");
                int data1 = sc.nextInt();
                insertAtStart(data1);
                break;
            case 3:
                System.out.println("Enter index to insert : ");
                int index = sc.nextInt();
                System.out.println("Enter element to insert : ");
                int data2 = sc.nextInt();
                insertAt(index, data2);
                break;
            case 4:
                System.out.println("Enter index to delete : ");
                int ind = sc.nextInt();
                deleteAt(ind);
                break;
            case 5:
                printList();
                break;
            case 6:
                break;
            default:
                System.out.println("Invalid choice");
            }
        }
        sc.close();
    }
}
```

Output :

```
D:\>javac CircularLinkedList.java

D:\>java CircularLinkedList
1. Add element at the end of the list
2. Insert element at the start of the list
3. Insert element at the given index
4. Delete element at the given index
5. Print List
6. Exit
```

```
Enter your choice :
1
Enter element to add :
10
1. Add element at the end of the list
2. Insert element at the start of the list
3. Insert element at the given index
4. Delete element at the given index
5. Print List
6. Exit
Enter your choice :
2
Enter element to insert :
20
1. Add element at the end of the list
2. Insert element at the start of the list
3. Insert element at the given index
4. Delete element at the given index
5. Print List
6. Exit
Enter your choice :
3
Enter index to insert :
2
Enter element to insert :
20
1. Add element at the end of the list
2. Insert element at the start of the list
3. Insert element at the given index
4. Delete element at the given index
5. Print List
6. Exit
Enter your choice :
4
Enter index to delete :
1
1. Add element at the end of the list
2. Insert element at the start of the list
```

```
3. Insert element at the given index
4. Delete element at the given index
5. Print List
6. Exit
Enter your choice :
5
20->20->...
1. Add element at the end of the list
2. Insert element at the start of the list
3. Insert element at the given index
4. Delete element at the given index
5. Print List
6. Exit
Enter your choice :
6

D:\>
```

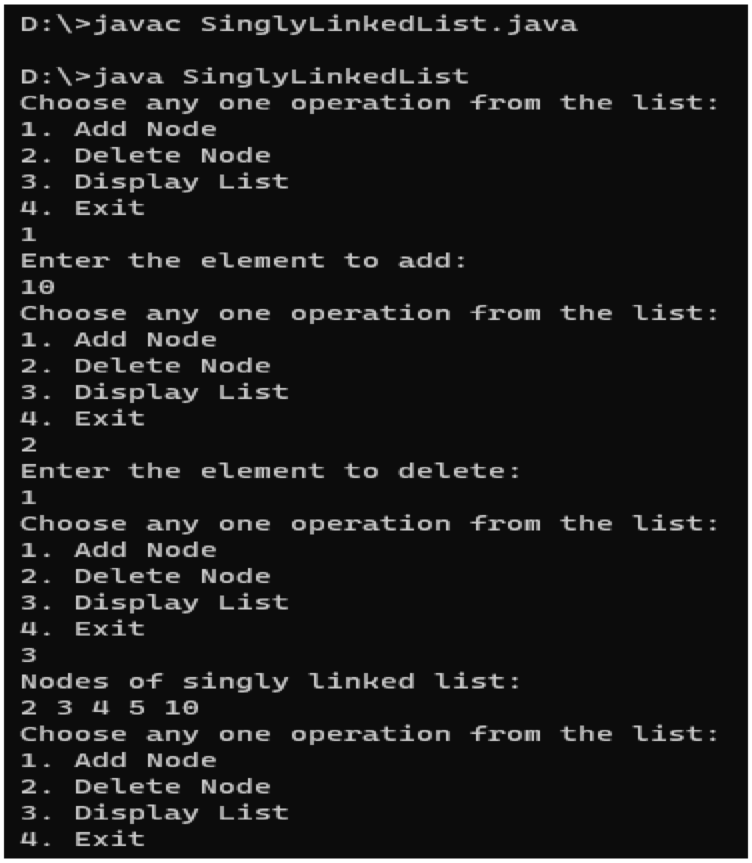
==

10.

```
import java.util.NoSuchElementException;
import java.util.*;

public class SinglyLinkedList {

    // Represents the node of list.
    class Node {
        int data;
        Node next;

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

    // Declaring head and tail pointer as null.
    public Node head = null;
    public Node tail = null;

    // This function will add the new node at the end of the list.
    public void addNode(int data) {
        // Create new node
        Node newNode = new Node(data);

        // Checks if the list is empty.
        if (head == null) {
            // If list is empty, both head and tail would point to new node.
            head = newNode;
            tail = newNode;
        } else {
            // tail will point to new node.
            tail.next = newNode;
            // New node will become new tail.
            tail = newNode;
        }
    }

    // This function will print all the node value of the list
    public void display()
    {
        // Node current will point to head
        Node current = head;

        if(head == null) {
            System.out.println("List is empty");
            return;
        }
        System.out.println("Nodes of singly linked list: ");
        while (current != null) {
            // Prints each node by incrementing pointer.
            System.out.print(current.data + " ");
            current = current.next;
        }
        System.out.println();
    }
```

```
//This function will delete a node based on the given value
public void deleteNode(int value) {
    // Start from head node
    Node current = head;
    Node prev = null;

    //Checks if the given value is present in the list
    if (current != null && current.data == value) {
        head = current.next;
        return;
    }

    // Search for the value to be deleted, keep track of the
    // previous node as we need to change temp.next
    while (current != null && current.data != value) {
        prev = current;
        current = current.next;
    }

    // If value was not present in linked list
    if (current == null)
        return;

    // Unlink the node from linked list
    prev.next = current.next;
}

public void operations(){
    System.out.println("Choose any one operation from the list:");
    System.out.println("1. Add Node");
    System.out.println("2. Delete Node");
    System.out.println("3. Display List");
    System.out.println("4. Exit");
    Scanner sc = new Scanner(System.in);
    int choice = Integer.parseInt(sc.nextLine());

    switch(choice){
            case 1:
                System.out.println("Enter the element to add:");
                int data = Integer.parseInt(sc.nextLine());
                addNode(data);
                break;
            case 2:
                System.out.println("Enter the element to delete:");
                int value = Integer.parseInt(sc.nextLine());
                deleteNode(value);
                break;
            case 3:
                display();
                break;
            case 4:
                System.exit(0);
                break;
            default:
                System.out.println("Please enter a valid option");
        }

        operations();
    }

    // Main Function
    public static void main(String[] args) {
        SinglyLinkedList list = new SinglyLinkedList();
        list.addNode(1);
        list.addNode(2);
        list.addNode(3);
        list.addNode(4);
        list.addNode(5);
        list.operations();
    }
}
```

Output:



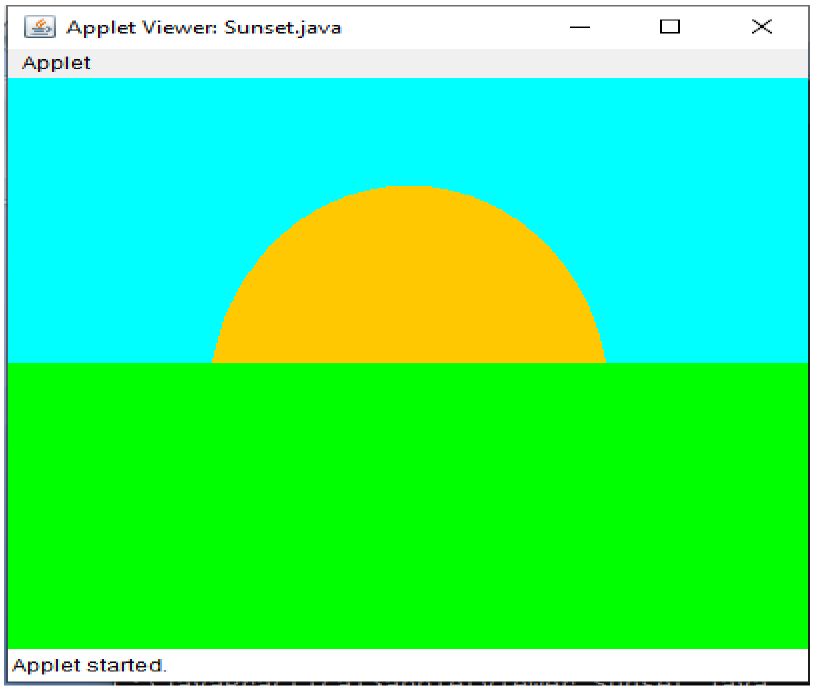
```
D:\>javac SinglyLinkedList.java

D:\>java SinglyLinkedList
Choose any one operation from the list:
1. Add Node
2. Delete Node
3. Display List
4. Exit
1
Enter the element to add:
10
Choose any one operation from the list:
1. Add Node
2. Delete Node
3. Display List
4. Exit
2
Enter the element to delete:
1
Choose any one operation from the list:
1. Add Node
2. Delete Node
3. Display List
4. Exit
3
Nodes of singly linked list:
2 3 4 5 10
Choose any one operation from the list:
1. Add Node
2. Delete Node
3. Display List
4. Exit
```

==

11.

```
/*
<applet code="Sunset.java" height=400 width=400></applet>
*/
import java.awt.*;
import java.applet.*;
public class Sunset extends Applet {

  public void paint(Graphics g) {

    // The sky
    g.setColor(Color.cyan);
    g.fillRect(0,0,400,200);

    // The sun
    g.setColor(Color.orange);
    g.fillOval(100,75,200,300);
    // The ground
    g.setColor(Color.green);
    g.fillRect(0,200,400,200);
  }
}
```

Output:



==

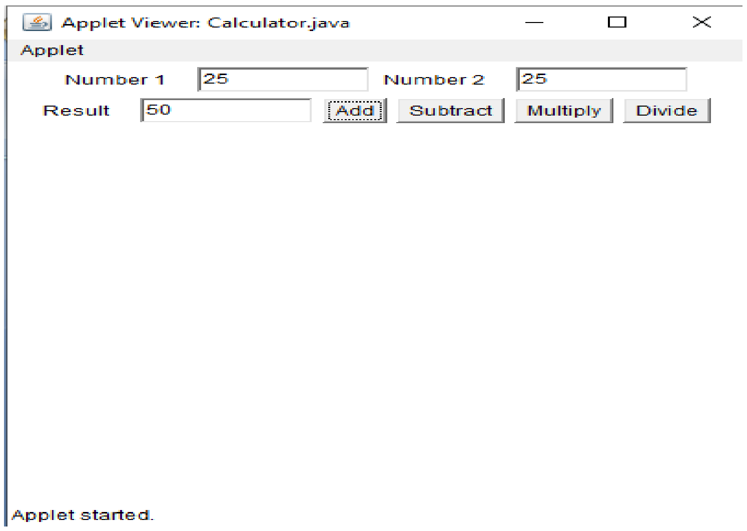
12.

```
import java.applet.Applet;
import java.awt.Button;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
/*
<applet code="Calculator.java" height=400 width=400>
</applet>
*/
public class Calculator extends Applet implements ActionListener
{
  TextField num1, num2, result;
  Button add, sub, mul, div;
```

```
public void init()
{
  Label l1 = new Label("Number 1");
  Label l2 = new Label("Number 2");
  Label l3 = new Label("Result");

  num1 = new TextField(10);
  num2 = new TextField(10);
  result = new TextField(10);

  add = new Button("Add");
  sub = new Button("Subtract");
  mul = new Button("Multiply");
  div = new Button("Divide");

  add(l1);
  add(num1);
  add(l2);
  add(num2);
  add(l3);
  add(result);
  add(add);
  add(sub);
  add(mul);
  add(div);

  add.addActionListener(this);
  sub.addActionListener(this);
  mul.addActionListener(this);
  div.addActionListener(this);
}

public void actionPerformed(ActionEvent e)
{
  String s1 = num1.getText();
  String s2 = num2.getText();
  int a = Integer.parseInt(s1);
  int b = Integer.parseInt(s2);
```

```
    int c = 0;

    if (e.getSource() == add) {
      c = a + b;
    } else if (e.getSource() == sub) {
      c = a - b;
    } else if (e.getSource() == mul) {
      c = a * b;
    } else if (e.getSource() == div) {
      c = a / b;
    }

    String res = String.valueOf(c);
    result.setText(res);
  }
}
```

Output:



```
25
25
50
Applet started.
```

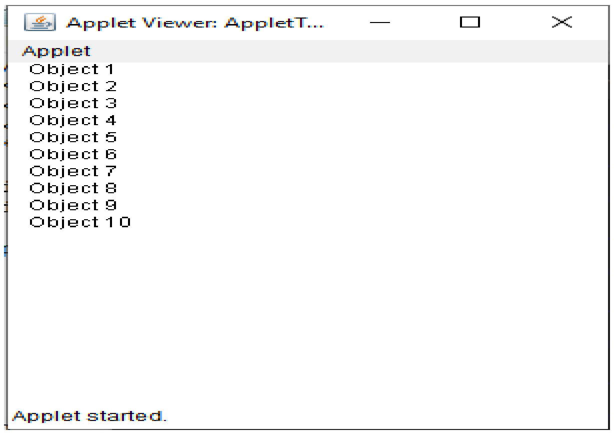
==

13.

```
/*
<applet code="AppletTag.java" width="300" height="300">
<param name="N" value="10" />
</applet>
*/
import java.applet.Applet;
import java.awt.*;
```

```
public class AppletTag extends Applet{

        public void paint(Graphics g){
                String n = getParameter("N");
                int num = Integer.parseInt(n);
                for(int i = 0; i < num; i++){
                        g.drawString("Object "+(i+1), 10, 10+(i*15));
                }
        }
}
```

Output :



```
Object 1
Object 2
Object 3
Object 4
Object 5
Object 6
Object 7
Object 8
Object 9
Object 10
```

==

14.

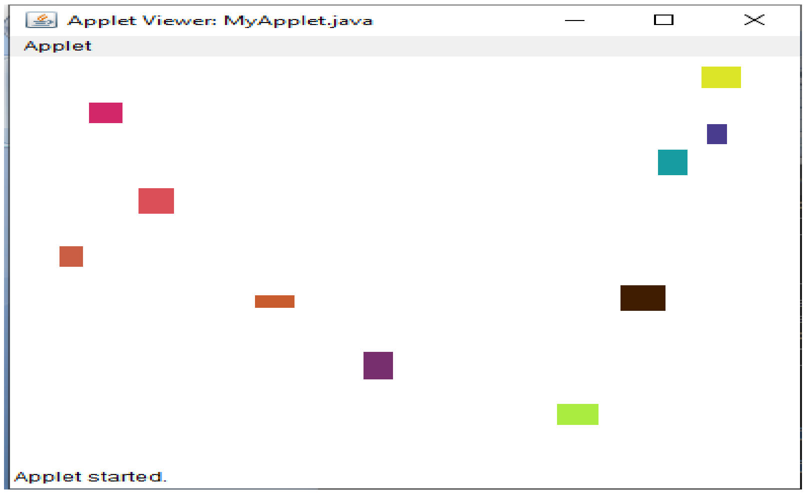
```
// This program demonstrates the creation and use of a package in Java
// Create a package "mypack"
package mypack;
// Declare a class
public class SampleClass {
  // Declare a method
  public void sayHello() {
    System.out.println("Hello from mypack");
  }
}
// Create another class to access the package
public class Demo {
  public static void main(String args[]) {
    SampleClass obj = new SampleClass();
    obj.sayHello();
  }
}
```

==

15.

```
import java.awt.Color;
import java.awt.Graphics;
/*
<applet code="MyApplet.java" height=400 width=400>
</applet>
*/
public class MyApplet extends java.applet.Applet {
        int number;
        public void init() {
                String numStr = getParameter("NUMBER");
                if (numStr == null) {
                        number = 10;
                } else {
                        number = Integer.parseInt(numStr);
                }
        }
        public void paint(Graphics g) {
                for (int i = 0; i < number; i++) {
                        int x = (int)(Math.random() * getSize().width);
                        int y = (int)(Math.random() * getSize().height);
                        int width = (int)(Math.random() * 20 + 10);
                        int height = (int)(Math.random() * 20 + 10);
                        int red = (int)(Math.random() * 256);
                        int green = (int)(Math.random() * 256);
                        int blue = (int)(Math.random() * 256);
                        Color color = new Color(red, green, blue);
                        g.setColor(color);
                        g.fillRect(x, y, width, height);
                }
        }
}
```

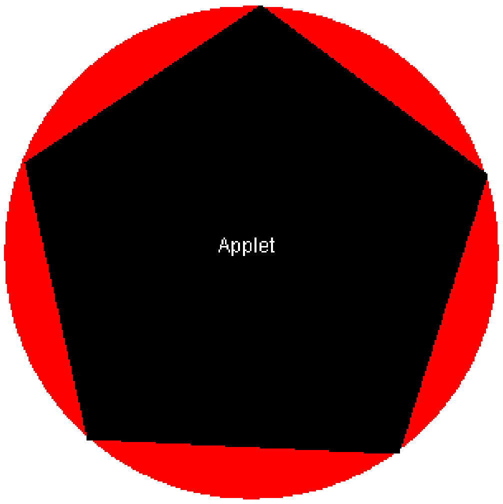
Output :

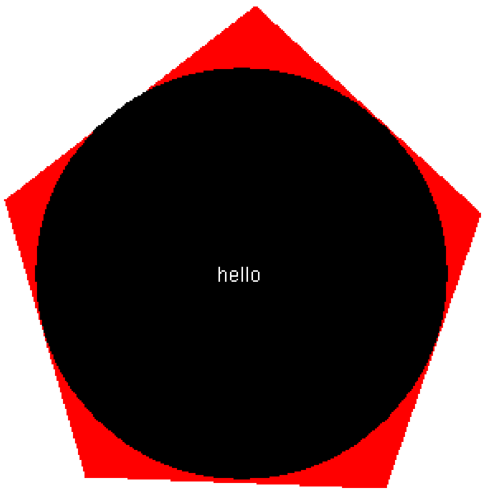


16.

```
Import java.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);} }
```

Output :


Applet


hello

==