Name : Italiya Sahil NareshBhai

Class : SYBCA

Div : 2

Roll No : 205

Subject : Java Programming Language(403)

1) Write a program for checking number is positive or negative.

```
import java.util.*;
public class Question{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Number : ");
        int rno = sc.nextInt();
        String value = rno<0?"You input number is nagative":"You input number is positive";
        System.out.println(value);
    }
}
```

2. Write a program to finding area of circle.

```
import java.util.*;
public class AreaOfCircle {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Radious to find area of circle :: ");
        double r = sc.nextDouble();
        double area =3.14*r*r;
        System.out.println(area);
    }
```

```
}
```

3) Write a program to find maximum and minimum number from given list.

```
import java.util.*;
 public class MaxAndMinOfList {
        public static void main(String[] args) {
        int[] array = {1,2,14,3,12,4,0,5};
        int max = array[0];
        int min = array[0];
        for (int i=0;i<array.length;i++){
                if(array[i] >max){
                        max = array[i];
                }
                if(array[i] < min){
                        min = array[i];
                }
        }
        System.out.println("Min value is "+min + "\nMax value is "+max);
        }
 }
```

4) Write a program to print following pattern.

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

public class FirstPattern {
        public static void main(String[] args) {
        for (int i=0;i<3;i++){
                for(int j=0;j<=i;j++){
```

```
                    System.out.print("*");
                }
                System.out.println(" ");
            }
        }
}
```

5) Write a program to print following pattern.

```
*   *   *
  *   *
    *
```

```
import java.util.*;
public class SecondPattern {
        public static void main(String[] args) {
        for (int i=1;i<=3;i++){
                for(int j=1;j<i;j++){
                        System.out.print(" ");
                }
                for(int j=i;j<=3;j++){
                        System.out.print("* ");
                }
                System.out.println(" ");
                }
          }
    }
```

6) Write a program to print multiplication table of given number.

```
    import java.util.*;
        public class MultiplicationTable {
                public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
```

```
            int r = sc.nextInt();
            for(int i=1; i<=10; i++){
                System.out.println(r+" X "+i+" = "+(r*i));
            }
        }
    }
```

7) Write a program to add two numbers using function overloading.

```
public class FunctionOverloading {
    static void add(int a,int b){
        System.out.println((a+b));
    }
    static void add(double a,double b){
        System.out.println((a+b));
    }
    public static void main(String[] args) {
    add(1,2);
    add(1.7,2.8);
    }
}
```

8) Write a program to input Employee Details and display it on proper format.

```
import java.util.*;

public class Employee_detail {
    static void display(int eid,String name){
    System.out.println("\tEid\tName");
    System.out.println("\t"+eid+"\t"+name);
    }
    public static void main(String[] args) {
    Employee e1 = new Employee();
```

```
        e1.input();
        display(e1.eid,e1.name);
        }
 }
 class Employee{
        int eid;
        String name;
        void input(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name :: ");
        name =sc.nextLine();
        System.out.print("Enter ID ::");
        eid = sc.nextInt();
        }
 }
```

9) Write a program to design three classes that accept dimension of triangle and rectangle and calculate area of rectangle and triangle.

```
 import java.util.*;

 class Dimention{
        public static void main(String[] args) {
        Triangle tr = new Triangle();
        Rectangle re = new Rectangle();
        tr.inputtr();
        re.inputre();
        }
 }
 class Triangle{
        double base;
```

```
        double height;
        void inputtr(){
        System.out.println(" :: Area Of Triangle :: ");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter value of base :");
        base = sc.nextDouble();
        System.out.print("Enter value of height :");
        height = sc.nextDouble();
        System.out.println("Area of the triangle is "+ (0.5*height*base));
        }
}
class Rectangle{
        double height;
        double width;
        void inputre(){
        System.out.println(" :: Area Of Rectangle :: ");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter value of height :");
        height = sc.nextDouble();
        System.out.print("Enter value of width :");
        width = sc.nextDouble();
        System.out.println("Area of the triangle is "+ (height*width));
        }
}
```

10) Write a program which design Bank Account class as Saving and Current Account and manage information accordingly.

```
class BankAccount{
        private String Name;
        private String Cid;
```

```
        protected Double BankBalance;
        public void displayBalance() {
        System.out.println("Current Balance: " + BankBalance);
    }
        public BankAccount(String Cid,String Name,Double BankBalance){
                this.Cid = Cid;
                this.Name = Name;
                this.BankBalance = BankBalance;
     }
        public void diposite(double amount){
                BankBalance += amount;
                displayBalance();
      }
        public void withdraw(double amount){
                if(amount < BankBalance){
                        BankBalance -= amount;
                        displayBalance();
                }
                else{
                        System.out.println("Withraw draw denied");
                }
         }
   }
   class SavingAccount extends BankAccount{
        private double interestRate;
        public SavingAccount(String Cid,String Name,Double BankBalance,Double
interestRate){
        super(Cid,Name,BankBalance);
        this.interestRate = interestRate;
    }
```

```
        public void calculateInterest(){
        double interest = BankBalance * (interestRate / 100);
        BankBalance += interest;
        System.out.println("Interest added: " + interest);
        displayBalance();
    }
    }
    class CurrentAccount extends BankAccount {
        private double overdraftLimit;
        public CurrentAccount(String Cid, String Name, double Bankdbalance, double
overdraftLimit) {
        super(Cid, Name, Bankdbalance);
        this.overdraftLimit = overdraftLimit;
        }
        @Override public void withdraw(double amount) {
                if (amount > (BankBalance + overdraftLimit)) {
                        System.out.println("Withdrawal denied. Overdraft limit exceeded.");
                }
                else {
                        BankBalance -= amount;
                        System.out.println("Withdrawn: " + amount);
                        displayBalance();
                }
          }
    }
    class Bank{
        public static void main(String args[]){
        SavingAccount savingsAccount = new SavingAccount("C111", "John Doe", 1000.0,
        5.0);
```

```
        CurrentAccount currentAccount = new CurrentAccount("C112", "Om Smith", 1500.0,
        500.0);
        System.out.println("\n\n Saving Account details");
        savingsAccount.diposite(500.0);
        savingsAccount.calculateInterest();
        savingsAccount.withdraw(200.0);

        //current Account
        System.out.println("\n\n Current Account details");
        currentAccount.diposite(500.0);
        currentAccount.withdraw(1500.0);
        }
    }
```

11) Write a program which design a class name Fan to represent fan properties according to these properties Fan operation will be performed.

```
class Fan {
    public static final int SLOW = 1;
    public static final int MEDIUM = 2;
    public static final int FAST = 3;
    private int speed;
    private boolean isOn;
    public Fan() {
            speed = SLOW;
            isOn = false;
    }
    public void turnOn() {
            isOn = true;
            System.out.println("Fan is turned on.");
            displayFanStatus();
    }
```

```
public void turnOff() {
        isOn = false;
        System.out.println("Fan is turned off.");
        displayFanStatus();
}
public void setSpeed(int newSpeed) {
        if (isOn && (newSpeed == SLOW || newSpeed == MEDIUM || newSpeed ==
        FAST)) {
        speed = newSpeed;
        System.out.println("Fan speed set to: " + getSpeedString());
        displayFanStatus();
}
        else {
                System.out.println("Invalid speed setting or fan is turned off.");
        }
}
public void displayFanStatus() {
        System.out.println("Fan Status - Speed: " + getSpeedString());
}
private String getSpeedString() {
switch (speed) {
        case SLOW:
                return "SLOW";
        case MEDIUM:
                return "MEDIUM";
        case FAST:
                return "FAST";
        default:
                return "UNKNOWN";
        }
```

```
        }
    }
    class FanApp {
        public static void main(String[] args) {
        Fan myFan = new Fan();
        myFan.turnOn();
        myFan.setSpeed(Fan.MEDIUM);
        myFan.turnOff();
        }
}
```

12) **Write a program that creates two interfaces 1. Direction 2. Drive Car. And creates two classes 1. DirectionBoard 2. Car which inherits above interfaces.**

```
        interface Direction{
                public void toLeft();
                public void toRight();
        }
        Interface DriveCar{
                public void start();
                public void stop();
        }
    class DirectionBoard implements Direction{
        public void toLeft(){
                System.out.println("Turn Left the car");
        }
        public void toRight(){
                System.out.println("Turn Right the car");
        }
    }
    class Car implements DriveCar{
```

```
        public void start(){
                System.out.println("Car is Start");
        }
        public void stop(){
                System.out.println("Car is Stop");
        }
    }
    class CarApp{
        public static void main(String args[]){
        DirectionBoard directionboard = new DirectionBoard();
        Car car = new Car();
        car.start();
        car.stop();

        //Direction board class
        System.out.println("\n\n direction Board class");
        directionboard.toLeft();
        directionboard.toRight();
        }
    }
```

13) Write a program to demonstrate partial implementation of interface and extending interfaces.

```
public class InterfaceDemo {
        public static void main(String[] args) {
        MyClass obj = new MyClass();
        obj.methodA();
        obj.methodB();
        obj.methodC();
        }
```

```
}
interface A {
    void methodA();
}
interface B extends A {
    void methodB();
}
interface C extends B {
    void methodC();
}
class MyClass implements C {
    @Override
    public void methodA() {
            System.out.println("Implementation of methodA");
    }
    @Override
    public void methodB() {
            System.out.println("Implementation of methodB");
    }
    @Override
    public void methodC() {
            System.out.println("Implementation of methodC");
    }
}
```

14) Write a program to accept 5 command line argument and then raise the custom exception if any argument is not from the list (“BCA”,”MCA”,”BBA”, ”MBA”,”OTHER”).

```
public class CustomExceptionExample {
    public static void main(String[] args) {
    if (args.length < 5) {
```

```
                System.out.println("Please provide 5 command line arguments.");
                return ;
        }
        try {
                validateArguments(args);
                System.out.println("All arguments are valid.");
        } catch (InvalidCourseException e) {
                System.out.println("Error: " + e.getMessage());
        }
        }
        private static void validateArguments(String[] args) throws      InvalidCourseException
{
        String[] validCourses = {"BCA", "MCA", "BBA", "MBA","OTHER"};
        for (String arg : args) {
                if (!isValidCourse(arg, validCourses)) {
                throw new InvalidCourseException(arg);
                }
        }
    }
        private static boolean isValidCourse(String course, String[]
        validCourses) {
        for (String validCourse : validCourses) {
        if (validCourse.equals(course)) {
                return true;
        }
        }
                return false;
        }
    }
    class InvalidCourseException extends Exception {
```

```
        public InvalidCourseException(String course) {
        super("Invalid course: " + course);
    }
   }
```

15) Write a program to demonstrate thow and throws keyword.

```
 import java.util.Scanner;
        public class ExceptionDemo {
        public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        try {
                validateNumber(number);
                System.out.println("Number is valid.");
        } catch (InvalidNumberException e) {
                System.out.println("Error: " + e.getMessage());
        }
 }
 private static void validateNumber(int number) throws
 InvalidNumberException {
        if (number <= 0) {
                throw new InvalidNumberException("Number must be greater than Zero”);
         }
        }
}
class InvalidNumberException extends Exception {
        public InvalidNumberException(String message) {
        super(message);
  }
```

```
}
```

16) Write a program to demonstrate custom exception called Nagetive Number Argument Exception.

```
import java.util.*;
class CustomException{
    public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    int age = sc.nextInt();
    try{
            if(age < 0){
            throw new NagativeNumberArgumentException("Nagetive
            number is not valid");
     }
 }
 catch(NagativeNumberArgumentException e){
            System.out.println(e.getMessage());
            }
        }
    }
    class NagativeNumberArgumentException extends Exception{
            public NagativeNumberArgumentException(String s){
            super(s);
    }
}
```

17) Write a program to input command line argument and display those string which start with ‘A’ and ‘a’.

```
import java.util.*;
class AStartedString{
    public static void main(String args[]){
```

```
            Scanner scn = new Scanner(System.in);
            System.out.print("Enter The String :: ");
            String str = scn.nextLine();
            String[] arr = str.split(" ");
            for(int i=0;i<arr.length;i++){
                if(arr[i].charAt(0) == 'A' || arr[i].charAt(0) == 'a'){
                    System.out.println(arr[i]);
                }
              }
          }
      }
```

18) Write java application which accept a string and display the string in reverse order by interchanging its odd positioned characters with even positioned characters.

```
 import java.util.*;
class ReverseString{
     public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter value");
    String value = sc.nextLine();
    String reverseString = reverse(value);
    System.out.println("revese string is "+reverseString);
}
public static String reverse(String value){
    if(value.length() == 0){
            System.out.println("String must contain more than one character");
            return "";
    }
    String temp = "";
    for(int i = value.length()-1; i>=0 ; i-=2){
```

```
        if (i!=0){
                temp += value.charAt(i-1) ;
        }
                 temp += value.charAt(i);
         }
        return temp;
        }
    }
```

19) Write a program to input two strings search similar characters from both string and replace it with '*'.

```
    import java.util.Scanner;
    public class ReplaceSimilarCharacters {
        public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter the first string: ");
        String firstString = sc.nextLine();
        System.out.print("Enter the second string: ");
        String secondString =sc.nextLine();
        String replacedFirstString = replaceSimilarCharacters(firstString, secondString);
        String replacedSecondString = replaceSimilarCharacters(secondString, firstString);
        System.out.println("First string with similar characters replaced: " +
        replacedFirstString);
        System.out.println("Second string with similar characters replaced: " +
        replacedSecondString);
        }
        private static String replaceSimilarCharacters(String original, String compare) {
                String result = "";
                for (int i = 0; i < original.length(); i++) {
                        char currentChar = original.charAt(i);
                        if (compare.indexOf(currentChar) != -1) {
```

```
                    result += "*";
                } else {
                    result += currentChar;
                }
            }
            return result;
        }
    }
```

20) Write a program to perform bubble sort on given inputted String.

```
class BubbleSort{
    public static void main(String args[]){
    int arr[] = {10,5,67,4,56,7,2,1,19};
    printarray(arr);

    bubbleSort(arr);

    printarray(arr);

}
    static void bubbleSort(int arr[]){
    int len = arr.length;
    for(int i = 0;i < len;i++){
        for(int j=0; j < len-i-1; j++){
            if(arr[j] > arr[j+1]){
            int temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
            }
        }
```

```
        }
    }
    static void printarray(int arr[]){
            int len = arr.length;
            for(int i = 0; i<len ; i++){
                System.out.print(arr[i]+ " ");
            }
            System.out.println(" \n\n");
        }
    }
```

21) Write an application that executes three threads from one thread class. One thread displays "JAVA" every 1 second. another display "PAPER" every 2 seconds and last one display "COLLEGE" every 3 seconds. Create the thread by using Runnable Interface.

```
class ThreadExample{
    public static void main(String[] args){
    Thread t1 = new Thread(new ThreadProg("JAVA",1));
    Thread t2 = new Thread(new ThreadProg("PAPER",2));
    Thread t3 = new Thread(new ThreadProg("COLLAGE",3));
    t1.start();
    t2.start();
    t3.start();
    }
}
class ThreadProg implements Runnable{
    int second;
    String value;
    public ThreadProg(String value,int second){
        this.value = value;
```

```
            this.second = second;
        }
        public void run(){
            for(int i=0;i<10;i++){
            try{
                Thread.sleep(1000*second);
                System.out.println(value);
            }catch(InterruptedException e){}
          }
       }
  }
```

22) Write a program to implement stack using thread.

```
import java.util.*;
public class ThreadSafeStack<T> {
    private Stack<T> stack;
    public ThreadSafeStack() {
    stack = new Stack();
    }
    public synchronized void push(T item) {
    stack.push(item);
    }
    public synchronized T pop() {
    if (stack.isEmpty()) {
            throw new EmptyStackException();
    }
    return stack.pop();
    }
    public synchronized boolean isEmpty() {
    return stack.isEmpty();
```

```
    }
    public static void main(String[] args) {
        ThreadSafeStack<Integer> stack = new ThreadSafeStack();
        Thread producerThread = new Thread(() -> {
        for (int i = 0; i < 10; i++) {
        stack.push(i);
        System.out.println("Pushed: " + i);
        try {
            Thread.sleep(100); // Simulate some work
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
});
    Thread consumerThread = new Thread(() -> {
    for (int i = 0; i < 10; i++) {
        int value = stack.pop();
        System.out.println("Popped: " + value);
        try {
            Thread.sleep(200); // Simulate some work
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
});
    producerThread.start();
    consumerThread.start();
    }
 }
```

23) **Write a java code which accepts name a of 10 students. Sort the names of students in ascending order. Display the names of students using thread class at interval of one seconds.**

```
import java.util.*;
class StudentThread extends Thread {
    private String[] names;
    public StudentThread(String[] names) {
        this.names = names;
    }
    @Override
    public void run() {
        for (String name : names) {
            System.out.println(name);
        try {
            Thread.sleep(1000); // Sleep for one second
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
      }
    }
    }
    public class StudentNameSort {
        public static void main(String[] args) {
        String[] names = new String[10];
        java.util.Scanner scanner = new java.util.Scanner(System.in);
        // Accept names of 10 students
        System.out.println("Enter names of 10 students:");
        for (int i = 0; i < 10; i++) {
            System.out.print("Student " + (i + 1) + ": ");
            names[i] = scanner.nextLine();
```

```
        }
        // Sort the names in ascending order
        Arrays.sort(names);
        // Create a thread to display student names
        StudentThread studentThread = new StudentThread(names);
        studentThread.start();
        }
    }
```

24) Write a program to demonstrate thread priorities.

```
public class ThreadPriorityDemo {
    public static void main(String[] args) {
    Thread thread1 = new Thread(new MyRunnable(), "Thread 1");
    Thread thread2 = new Thread(new MyRunnable(), "Thread 2");
     Thread thread3 = new Thread(new MyRunnable(), "Thread 3");
    // Set priorities for threads
    thread1.setPriority(Thread.MIN_PRIORITY); // 1
    thread2.setPriority(Thread.NORM_PRIORITY); // 5
    thread3.setPriority(Thread.MAX_PRIORITY); // 10
    // Start threads
    thread1.start();
    thread2.start();
    thread3.start();
    }
    static class MyRunnable implements Runnable {
    @Override
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(Thread.currentThread().getName() + " - Priority: "
        +
```

```
                Thread.currentThread().getPriority() + " - Count: " + i);
                    try {
                            Thread.sleep(1000); // Sleep for 1 second
                    } catch (InterruptedException e) {
                            e.printStackTrace();
                    }
                }
            }
        }
```

25) Write a program to demonstrate thread synchronization.

```
public class ThreadSynchronizationDemo {
    public static void main(String[] args) throws InterruptedException {
    SharedResource sharedResource = new SharedResource();
    IncrementThread incrementThread = new IncrementThread(sharedResource);
    incrementThread.start();
     DecrementThread decrementThread = new   DecrementThread(sharedResource);
    decrementThread.start();
    incrementThread.join();
    decrementThread.join();
    System.out.println("Count: " + sharedResource.getCount());
    }
}
class SharedResource {
    private int count = 0;
    public synchronized void increment() {
            count++;
    }
    public synchronized void decrement() {
```

```
                count--;
        }
        public int getCount() {
                return count;
        }
  }
  class IncrementThread extends Thread {
        private SharedResource sharedResource;
        public IncrementThread(SharedResource sharedResource) {
                this.sharedResource = sharedResource;
        }
        @Override
         public void run() {
        for (int i = 0; i < 1000; i++) {
                sharedResource.increment();
        }
   }
}
class DecrementThread extends Thread {
        private SharedResource sharedResource;
        public DecrementThread(SharedResource sharedResource) {
                this.sharedResource = sharedResource;
        }
        @Override
        public void run() {
                for (int i = 0; i < 1000; i++) {
                        sharedResource.decrement();
                }
        }
```

```
}
```

26) Write a java application which displays a Hexagon containing a circle within the pentagon where the circumference of the circle touches to the edges of the pentagon. Provide separate colors to both the objects. Display your name in center position of the circle.

```
import javax.swing.*;
import java.awt.*;
public class HexagonWithCircle extends JPanel {
    private static final int WIDTH = 600;
    private static final int HEIGHT = 600;
    public HexagonWithCircle() {
    setPreferredSize(new Dimension(WIDTH, HEIGHT));
    }
    @Override
    protected void paintComponent(Graphics g) {
            super.paintComponent(g);
    // Set colors
    g.setColor(Color.BLUE); // Color for hexagon
    g.fillRect(0, 0, WIDTH, HEIGHT); // Background color
    g.setColor(Color.RED); // Color for pentagon
    int[] xPentagon = {300, 500, 500, 400, 300};
    int[] yPentagon = {150, 200, 400, 500, 450};
    g.fillPolygon(xPentagon, yPentagon, 5); // Draw pentagon
    g.setColor(Color.YELLOW); // Color for circle
    int radius = 50;
    int centerX = 400; // X coordinate of center of the circle
    int centerY = 325; // Y coordinate of center of the circle
    g.fillOval(centerX - radius, centerY - radius, 2 * radius, 2 * radius);
    g.setColor(Color.BLACK); // Color for text
```

```
        g.setFont(new Font("Arial", Font.BOLD, 16)); // Set font for text
        String name = "Your Name"; // Your name
        FontMetrics fm = g.getFontMetrics();
        int nameWidth = fm.stringWidth(name);
        int nameHeight = fm.getHeight();
        g.drawString(name, centerX - nameWidth / 2, centerY + nameHeight / 4);
        }
        public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
        JFrame frame = new JFrame("Hexagon with Circle");
        Frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().add(new HexagonWithCircle(), BorderLayout.CENTER);
        frame.pack();
        frame.setLocationRelativeTo(null); // Center window on screen
        frame.setVisible(true);
        });
    }
}
```

27) Write a javacode (application or applet) which display current Date and Time Use thread class to execute the code.

```
import java.util.Date;

public class DateTimeDisplay {
    public static void main(String[] args) {
    DateTimeThread dateTimeThread = new DateTimeThread();
    dateTimeThread.start();
    }
}
class DateTimeThread extends Thread {
```

```
        @Override
        public void run() {
                while (true) {
                  Date currentDate = new Date();
                System.out.println("Current Date and Time: " + currentDate);
                try {
                        Thread.sleep(1000); // Sleep for 1 second
                } catch (InterruptedException e) {
                        e.printStackTrace();
                }
            }
       }
  }
```

28) **Write an applet for moving banner.**

```
 import java.applet.Applet;
 import java.awt.Color;
 import java.awt.Font;
 import java.awt.Graphics;
 import java.awt.event.ActionEvent;
 import java.awt.event.ActionListener;
public class MovingBannerApplet extends Applet implements
        ActionListener, Runnable {
        private String message = "Moving Banner";
        private int xCoordinate = 10;
        private int yCoordinate = 50;
        private Thread thread;
        private boolean isRunning = false;
        public void init() {
                setBackground(Color.white);
```

```
        setForeground(Color.red);
        setFont(new Font("Arial", Font.BOLD, 20));
        thread = new Thread(this);
        thread.start();
        }
        public void actionPerformed(ActionEvent e) {
        }
        public void run() {
        while (true) {
                xCoordinate += 5;
                if (xCoordinate > getWidth()) {
                        xCoordinate = -100;
                }
                repaint();
                try {
                        Thread.sleep(100);
                } catch (InterruptedException e) {
                        e.printStackTrace();
                }
            }
        }
        public void paint(Graphics g) {
                g.drawString(message, xCoordinate, yCoordinate);
        }
    }
```

29) Write an applet for bouncing Ball using thread.

```
 import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
```

```
public class BouncingBallApplet extends Applet implements Runnable {
    private int x = 150;
     private int y = 50;
     private int xSpeed = 2;
    private int ySpeed = 3;
    private int radius = 20;
    private Thread thread;
    public void init() {
            setBackground(Color.white);
    }
    public void start() {
            thread = new Thread(this);
            thread.start();
    }
    public void stop() {
            thread.interrupt();
            thread = null;
    }
    public void run() {
            while (true) {
                    x += xSpeed;
                    y += ySpeed;
            if (x - radius < 0 || x + radius > getWidth()) {
                     xSpeed = -xSpeed;
            }
            if (y - radius < 0 || y + radius > getHeight()) {
                    ySpeed = -ySpeed;
            }
            repaint();
```

```
                try {
                        Thread.sleep(16);
                } catch (InterruptedException e) {
                        break;
                }
            }
        }
        public void paint(Graphics g) {
                g.setColor(Color.red);
                g.fillOval(x - radius, y - radius, 2 * radius, 2 * radius);
        }
}
```

30) Write a program to demonstrate communication between two applet using ActionListner. Applet 1 Contain login page Applet 2 Diaply Welcome message for user who is logged in Applet1.

```
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;
    public class LoginApplet extends Applet implements ActionListener {
    TextField usernameField;
    Button loginButton;
    public void init() {
            Label usernameLabel = new Label("Username:");
            usernameField = new TextField(20);
            loginButton = new Button("Login");
            loginButton.addActionListener(this);
            add(usernameLabel);
```

```
            add(usernameField);
            add(loginButton);
        }
        public void actionPerformed(ActionEvent e) {
            String username = usernameField.getText();
            Applet2 applet2 = (Applet2) getAppletContext().getApplet("Applet2");
            applet2.displayWelcomeMessage(username);
        }
    }
    //second applet
    import java.applet.Applet;
    import java.awt.Label;
        public class WelcomeApplet extends Applet {
        Label welcomeLabel;
        public void init() {
            welcomeLabel = new Label("");
            add(welcomeLabel);
        }
        public void displayWelcomeMessage(String username) {
        welcomeLabel.setText("Welcome, " + username + "!");
        }
    }
    //write both applet in html code
   <html>
        <head>
        <title>Communication Between Applets</title>
        </head>
        <body>
        <applet code="LoginApplet.class" width="300" height="200"></applet>
```

```
        <applet code="WelcomeApplet.class" width="300" height="200">
        </applet>
        </body>
    </html>
```

31) Write a program to demonstrate Applet HTML Tag.

```
    //java applet program
    import java.applet.Applet;
    import java.awt.Graphics;
    public class MyApplet extends Applet {
        public void paint(Graphics g) {
        g.drawString("Hello, World!", 20, 20);
        }
    }
    //Applet Html Tag program
    <!DOCTYPE html>
    <html>
        <head>
        <title>Applet HTML Tag Demo</title>
        </head>
        <body>
        <h1>Applet HTML Tag Demo</h1>
        <hr>
        <p>This is a simple demonstration of using the &lt;applet&gt; HTML tag to embed
        an applet in a web page.</p>
        <applet code="MyApplet.class" width="200" height="100">
        Your browser does not support Java applets.
        </applet>
        <hr>
        <p>End of demonstration.</p>
```

```
    </body>
</html>
```

32) Write an applet program that gets number of rectangles from the user using()and draw the rectangle in different position.

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.util.Scanner;
public class RectangleApplet extends Applet {
private int numRectangles;
public void init() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter the number of rectangles: ");
    numRectangles = scanner.nextInt();
    scanner.close();
    }
    public void paint(Graphics g) {
        int width = getWidth();
        int height = getHeight();
        int rectWidth = 50;
        int rectHeight = 30;
        // Set different colors for each rectangle
        Color[] colors = {Color.RED, Color.GREEN, Color.BLUE, Color.YELLOW,
        Color.ORANGE};
        for (int i = 0; i < numRectangles && i < colors.length; i++) {
            g.setColor(colors[i]);
            int x = (int) (Math.random() * (width - rectWidth));
            int y = (int) (Math.random() * (height - rectHeight));
            g.fillRect(x, y, rectWidth, rectHeight);
```

```
            }
        }
    }
      //HTML file
    <!DOCTYPE html>
    <html>
    <head>
    <title>Rectangle Applet</title>
    </head>
    <body>
    <applet code="RectangleApplet.class" width="400" height="400">
            Your browser does not support Java applets.
    </applet>
    </body>
    </html>
```

33) Write a program add/sub/mul/div of 2 numbers in Applet. Accept value in 2 textboxes & create button name “+”,”-“,”/”,”*”, when you click them, it will give result in third box.

```
//Applet program
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;
public class CalculatorApplet extends Applet implements ActionListener {
    private TextField numField1, numField2, resultField;
    private Button addButton, subtractButton, multiplyButton, divideButton;
    public void init() {
            Label numLabel1 = new Label("Number 1:");
```

```
        Label numLabel2 = new Label("Number 2:");
        Label resultLabel = new Label("Result:");
        numField1 = new TextField(10);
        numField2 = new TextField(10);
        resultField = new TextField(10);
        resultField.setEditable(false); // Result field is not editable by the   user
        addButton = new Button("+");
        subtractButton = new Button("-");
        multiplyButton = new Button("*");
        divideButton = new Button("/");
        // Add action listeners to buttons
        addButton.addActionListener(this);
        subtractButton.addActionListener(this);
        multiplyButton.addActionListener(this);
        divideButton.addActionListener(this);
        add(numLabel1);
        add(numField1);
        add(numLabel2);
        add(numField2);
        add(addButton);
        add(subtractButton);
        add(multiplyButton);
        add(divideButton);
        add(resultLabel);
        add(resultField);
    }
    public void actionPerformed(ActionEvent e) {
        double num1 = Double.parseDouble(numField1.getText());
        double num2 = Double.parseDouble(numField2.getText());
```

```
            double result = 0.0;
            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) {
                if (num2 != 0) {
                    result = num1 / num2;
                } else {
                    resultField.setText("Error: Division by zero");
                    return;
                }
            }
            resultField.setText(String.valueOf(result));
        }
    }
//Html program
<!DOCTYPE html>
<html>
<head>
<title>Calculator Applet</title>
</head>
<body>
<applet code="CalculatorApplet.class" width="400" height="200">
            Your browser does not support Java applets.
</applet>
```

```
</body>
</html>
```

34) **Write an applet to run analog clock with current date & time in following format.**

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Point;
import java.text.SimpleDateFormat;
import java.util.Date;
public class AnalogClockApplet extends Applet implements Runnable {
    private Thread thread;
    public void init() {
        setBackground(Color.white);
    }
    public void start() {
        If (thread == null) {
            thread = new Thread(this);
            thread.start();
        }
    }
    public void stop() {
        if (thread != null) {
            thread.interrupt();
            thread = null;
        }
    }
    public void run() {
```

```
while (true) {
        try {
                Thread.sleep(1000); // Update every second
                repaint(); // Request to repaint the applet
        } catch (InterruptedException e) {
                break;
        }
    }
}
public void paint(Graphics g) {
// Get current time
Date now = new Date();
SimpleDateFormat dateFormat = new SimpleDateFormat("EEE, MMM dd, yyyy
HH:mm:ss");
String currentTime = dateFormat.format(now);
// Draw the clock
int centerX = getWidth() / 2;
int centerY = getHeight() / 2;
int radius = Math.min(centerX, centerY) - 10;
 // Draw clock face
g.setColor(Color.black);
g.drawOval(centerX - radius, centerY - radius, 2 * radius, 2 * radius);
// Draw hour, minute, and second hands
g.setColor(Color.blue);
drawHand(g, centerX, centerY, radius * 0.5, 30 * now.getHours() + now.getMinutes()
/ 2);
g.setColor(Color.green);
drawHand(g, centerX, centerY, radius * 0.7, 6 * now.getMinutes() + now.getSeconds()
/ 10);
g.setColor(Color.red);
```

```
        drawHand(g, centerX, centerY, radius * 0.9, 6 * now.getSeconds());
        // Draw current date and time
        g.setColor(Color.black);
        g.setFont(new Font("Arial", Font.BOLD, 12));
        g.drawString(currentTime, 10, getHeight() - 10);
        }
         private void drawHand(Graphics g, int x, int y, double length, int angle)   {
                angle -= 90;
                double radian = Math.toRadians(angle);
                int x2 = x + (int) (length * Math.cos(radian));
                int y2 = y + (int) (length * Math.sin(radian));
                g.drawLine(x, y, x2, y2);
        }
 }
 //Html Program
<!DOCTYPE html>
<html>
<head>
<title>Analog Clock Applet</title>
</head>
<body>
<applet code="AnalogClockApplet.class" width="300" height="300">
                Your browser does not support Java applets.
</applet>
</body>
</html>
```

35) Write an applet program to print name in following pictures & fill different colors in picture.

```
import java.applet.Applet;
```

```
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
public class PictureApplet extends Applet {
    public void paint(Graphics g) {
            setBackground(Color.white);
            // Draw the big circle containing a pentagon
            g.setColor(Color.blue);
            g.fillOval(50, 50, 300, 300); // Big circle
            g.setColor(Color.yellow);
            int[] xPentagon = {200, 250, 300, 250, 200};
            int[] yPentagon = {100, 50, 100, 150, 100};
            g.fillPolygon(xPentagon, yPentagon, 5); // Pentagon
            g.setColor(Color.black);
            g.drawString("APPLET", 180, 180);
            g.setColor(Color.green);
            int[] xPentagon2 = {400, 450, 500, 450, 400};
            int[] yPentagon2 = {100, 50, 100, 150, 100};
            g.fillPolygon(xPentagon2, yPentagon2, 5);
            g.setColor(Color.red);
            g.fillOval(425, 75, 50, 50); // Circle inside pentagon
            g.setColor(Color.black);
            g.drawString("HELLO", 410, 120); // Text "HELLO" in the center
            }
    }
    //HTML code
    <!DOCTYPE html>
    <html>
    <head>
```

```
<title>Picture Applet</title>
</head>
<body>
<applet code="PictureApplet.class" width="600" height="400">
        Your browser does not support Java applets.
</applet>
</body>
</html>
```

36) Write a package that shows all combinations of the access control modifiers.

```
package mypackage;
    public class MyClass {
    public int publicVar;
    protected int protectedVar;
    int defaultVar;
    private int privateVar;
    public void publicMethod() {}
    protected void protectedMethod() {}
    void defaultMethod() {}
        private void privateMethod() {}
    }
```