JAVA PRACTICAL ANS

1. Write a program lo display ASCII value of a number 9?

ANS:

```
public class Demo {
   public static void main(String[] args) {
      char c='9';
      int asciiValue=(int)c;
      System.out.println("ASCII VALUE OF NUMBER "+c+" IS "+asciiValue);
   }
}
```

2. Write a program which displays functioning of ATM machine, (Hint: Withdraw, Deposit, Check Balance and Exit).

ANS:

```
Import java.util.*;

public class Demo {
  float balance=0;
  Scanner sc=new Scanner(System.in);
  public void deposit(){
     System.out.println("ENTER AMOUNT TO ADD: ");
     float addAmount=sc.nextFloat();
     if (addAmount<1) {
        System.out.println("SORRY CANNOT ENTER NEGATIVE AMOUNT! \n");
     }else{
        balance=balance+addAmount;
        System.out.println("AMOUNT ADDED SUCCESSFULLY \n");
     }
  }
  public void checkBalance(){
     System.out.println("YOUR BALANCE IS $ "+balance+"\n");
  }
  public void withdraw(){
     System.out.println("ENTER AMOUNT TO WITHDRAW: ");
     float deductamount=sc.nextFloat();
     if (deductamount>balance) {
        System.out.println("INSUFFICIENT BALANCE! \n");
     }else{
        balance=balance-deductamount;
        System.out.println("AMOUNT WITHDRAWN SUCCESSFULLY \n");
     }
  }
  public static void main(String[] args) {
     Demo d=new Demo();
     int choice;
     Scanner sc=new Scanner(System.in);
     do{
```

```
            System.out.println("CHOOSE ONE:\n1.DEPOSIT\n2.WITHDRAW\n3.CHECK
    BALANCE\n4.EXIT");
            choice=sc.nextInt();
            switch(choice){
               case 1 :
                  d.deposit();
                  break;
               case 2 :
                  d.withdraw();
                  break;
               case 3 :
                  d.checkBalance();
                  break;
               case 4:
                  System.exit(0);
                  break;
               default :
                  System.out.println("INVALID CHOICE!");
            }
            System.out.println("\nDO YOU WANT CONTINUE? PRESS 1 FOR YES OR OTHER KEY
    FOR NO \n");
            choice=sc.nextInt();
         }while(choice==1);
      }
   }
```

3. Write a program to print all the Armstrong numbers from 0 to 999?

ANS:

```
Import java.util.*;

public class Demo {
   public static void main(String[] args) {
      System.out.println("Armstrong numbers from 0 to 999:");
      // Iterate through numbers from 0 to 999
      for (int i = 0; i <= 999; i++) {
         if (isArmstrong(i)) {
            System.out.println(i);
         }
      }
   }

   public static boolean isArmstrong(int num){
        int originalnum=num,n=0,d,res=0;
        while (originalnum!=0) {
             originalnum = originalnum/10;
             n++;
        }
        originalnum=num;
        while (originalnum!=0) {
            d=originalnum%10;
```

```
            res+=(int)Math.pow(d,n);
            originalnum=originalnum/10;
        }
        return res==num;
    }
}
```

4. Write a program to check whether the given number is prime or not.

ANS:

```
import java.util.*;

class PrimeNum{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter a number: ");
        int no=sc.nextInt();
        int count=0;
        System.out.println("Your Entered number is: " +no);
        if(no>1){
        for(int i=1;i<=no;i++){
            if(no%i==0){
              count++;
            }
          }
          if(count==2){
             System.out.println(no+ " is a Prime number ");
          }else{
             System.out.println(no+ " is not a Prime number ");
          }
        }else{
          System.out.println(no+ " is not a Prime number ");
        }
    }
}
```

5. Write a program to find reverse of a number.

ANS:

```
import java.util.*;

class reverseOfDigit{
public static void main(String args[]){
Scanner sc=new Scanner(System.in);
System.out.println("ENTER YOUR NUMBER: ");
int n=sc.nextInt();
System.out.print("REVERSE OF NUMBERS OF DIGIT "+n);
int rev=0,d;
while(n!=0){
d=n%10;
```

```
rev=(rev*10)+d;
n=n/10;
}
System.out.println(" IS: "+rev);
}
}
```

6. Write a Java Program to find out the even numbers from 1 to 100 using for loop.

ANS:

```
public class EvenFind {
  public static void main(String[] args) {
    System.out.println("Armstrong numbers from 0 to 999:");
    // Iterate through numbers from 0 to 999
    for (int i = 1; i <= 100; i++) {
      if (i%2==0) {
        System.out.println(i);
      }
    }
  }
}
```

7. Write a program to sort the elements of an array in ascending order.

ANS:

```
public class arraySort {
  public static void main(String[] args) {
    int arr[]={20,30,40,10,7};
    int i,j,n=arr.length,temp;
    System.out.print("Array before sorting: ");
    for (i = 0; i < arr.length; i++) {
        System.out.print(arr[i]+" ");
    }
    System.out.println();
    for (i = 0; i < n; i++) {
      for(j=0;j<n-i-1;j++){
        if (arr[j]>arr[j+1]) {
          temp=arr[j];
          arr[j]=arr[j+1];
          arr[j+1]=temp;
        }
      }
    }
    System.out.print("Array after sorting: ");
    for (i = 0; i < arr.length; i++) {
        System.out.print(arr[i]+" ");
    }
  }
}
```

8. Write a program to show the use of copy constructor.

ANS:

```
class Student {
```

```
    private String name;
    private int age;

    // Constructor
    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Copy constructor
    public Student(Student anotherStudent) {
        this.name = anotherStudent.name;
        this.age = anotherStudent.age;
    }

    // Method to display student information
    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

public class CopyConstructorExample {
    public static void main(String[] args) {
        // Creating a student object using the constructor
        Student originalStudent = new Student("Alice", 20);

        // Displaying original student information
        System.out.println("Original Student:");
        originalStudent.display();
        System.out.println();

        // Creating a copy of the original student using the copy constructor
        Student copiedStudent = new Student(originalStudent);

        // Displaying copied student information
        System.out.println("Copied Student:");
        copiedStudent.display();
    }
}
```

9. Write a program to print the sum, difference and product of two complex numbers by creating a class named "Complex" with separate methods for each operation whose real and imaginary parts are entered by user?

ANS:

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

```
class Complex{
   int real,img;
   Scanner sc=new Scanner(System.in);
   public void setValues(){
      System.out.println("Enter the Real part:");
      real =sc.nextInt();
      System.out.println("Enter the Imaginary part:");
      img =sc.nextInt();
   }
    public void add(Complex c1,Complex c2)  {
      real=c1.real+c2.real;
      img=c1.img+c2.img;
      System.out.println("Sum of two complex numbers is : "+"( i" +real+" +i"+img+")");
    }
    public void sub(Complex c1,Complex c2)  {
      real=c1.real-c2.real;
      img=c1.img-c2.img;
      System.out.println("DIFFFERENCE of two complex numbers: "+"( i" +real+" +i"+img+")");
    }
    public void mul(Complex c1,Complex c2)  {
      real=c1.real * c2.real - c1.img * c2.img;
      img=c1.real * c2.img + c1.img * c2.real;
      System.out.println("PRODUCT of two complex numbers: "+"(" +real+" +i"+img+")");
    }
    public void getVAlues(){
      System.out.println(" complex number: "+"(" +real+" +i"+img+")");
   }
}

public class Demo {
   public static void main(String[] args) {
      Complex c1= new Complex(),c2=new Complex(),c3=new Complex();
      System.out.println("First Complex Number ");
      c1.setValues();
      System.out.println("\nSecond Complex Number ");
      c2.setValues();
      System.out.println("\nDisplaying the values of both complex numbers:");
      c1.getVAlues();
      c2.getVAlues();
      c3.add(c1,c2);
      c3.sub(c1,c2);
      c3.mul(c1,c2);
   }
}
```

Output:-

```
First Complex Number
Enter the Real part:
2
Enter the Imaginary part:
3

Second Complex Number
Enter the Real part:
4
Enter the Imaginary part:
-5

Displaying the values of both complex numbers:
 complex number: (2 +i3)
 complex number: (4 +i-5)
Sum of two complex numbers is : ( i6 +i-2)
DIFFFERENCE of two complex numbers: ( i-2 +i8)
PRODUCT of two complex numbers: (23 +i2)
```

10. Write a program to copy all elements of one array into another array.

ANS:

```
public class CopyArray {
   public static void main(String[] args) {
      int arr[]={20,30,40,10,7};
      int copyarr[]=new int[arr.length];
      int i,j,n=arr.length,temp;
      System.out.println("Array before copying: ");
      System.out.println("ORIGINAL ARRAY: ");
      for (i = 0; i < arr.length; i++) {
            System.out.print(arr[i]+" ");
      }
      System.out.println("\nCOPIED ARRAY: ");
      for (i = 0; i < arr.length; i++) {
         System.out.print(copyarr[i]+" ");
      }
      System.out.println();

      for (i = 0; i < n; i++) {
         copyarr[i]=arr[i];
      }
      System.out.println("Array after copying: ");
      System.out.println("ORIGINAL ARRAY: ");
      for (i = 0; i < arr.length; i++) {
            System.out.print(arr[i]+" ");
      }
      System.out.println("\nCOPIED ARRAY: ");
      for (i = 0; i < arr.length; i++) {
         System.out.print(copyarr[i]+" ");
      }
   }
}
```

11. Define a class employee with data members 'empid, name and salary'. Accept data for three objects and display it.

ANS:

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

```
class employee{
 int id;
 String name;
 float salary;
 void setEmployee(int i,String n,float sal) {
 this.id=i;
 this.name=n;
 this.salary=sal;
 }
 void getEmployee() {
 System.out.println("EMPLOYEE NAME: "+name);
 System.out.println("EMPLOYEE ID: "+id);
 System.out.println("EMPLOYEE SALARY: "+salary);
 }
}
public class Demo {
 public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    employee e[]= new employee[3];
    for (int i=0 ; i<e.length ; i++) {
     e[i] = new employee();
    }
    e[0].setEmployee(10, "aditya", 20000);
    e[1].setEmployee(11, "siddharth", 5000);
    e[2].setEmployee(12, "shrushti", 5000);
    System.out.println("\n\nDisplaying the Details of all Employees are as follows:\n");
    for (int j=0 ; j<e.length ; j++) {
     System.out.println("\nDetails of Employee - "+(j+1)+" \n");
     e[j].getEmployee();
    }
 }
}
```

12. Write a program to add 2 integers, 2 string and 2 float values in a vector. Remove the element specified by the user and display the list.

ANS:

```
import java.util.*;

public class Demo {
 public static void main(String[] args){
    Vector v=new Vector(10);
    Integer i1=new Integer(10);
    Integer i2=new Integer(20);
    String s1=new String("ADITYA");
    String s2=new String("SIDDHARTH");
    float f1=new Float(3.14);
    float f2=new Float(5.13);
    System.out.println("VECTOR BEFORE ADDING ELEMENTS: "+v);
    v.addElement(i1);
    v.addElement(i2);
```

```
    v.addElement(s1);
    v.addElement(s2);
    v.addElement(f1);
    v.addElement(f2);
    System.out.println("\nVECTOR AFTER ADDING ELEMENTS: "+v);
  }
}
```

13. Write a program to check whether the string provided by the user is palindrome or not?

ANS:

```
import java.util.Scanner;

public class PalindromeChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a string
        System.out.println("Enter a string to check if it's a palindrome:");
        String input = scanner.nextLine();

        // Check if the input string is a palindrome
        boolean isPalindrome = true;
        int left = 0;
        int right = input.length() - 1;

        // Continue until left pointer is less than or equal to right pointer
        while (left < right) {
            // If characters at current positions are not equal, it's not a palindrome
            if (input.charAt(left) != input.charAt(right)) {
                isPalindrome = false;
                break; // No need to continue checking
            }
            // Move left pointer to the right and right pointer to the left
            left++;
            right--;
        }

        // Display the result
        if (isPalindrome) {
            System.out.println("The string \"" + input + "\" is a palindrome.");
        } else {
            System.out.println("The string \"" + input + "\" is not a palindrome.");
        }

        scanner.close();
    }
}
```

14. Write a java program to sort a 1-d array in ascending order using bubble-sort.

ANS:

```
public class arraySort {
   public static void main(String[] args) {
      int arr[]={20,30,40,10,7};
      int i,j,n=arr.length,temp;
      System.out.print("Array before sorting: ");
      for (i = 0; i < arr.length; i++) {
            System.out.print(arr[i]+" ");
      }
      System.out.println();
      for (i = 0; i < n; i++) {
         for(j=0;j<n-i-1;j++){
            if (arr[j]>arr[j+1]) {
               temp=arr[j];
               arr[j]=arr[j+1];
               arr[j+1]=temp;
            }
         }
      }
      System.out.print("Array after sorting: ");
      for (i = 0; i < arr.length; i++) {
            System.out.print(arr[i]+" ");
      }
   }
}
```

15. Write a program to show the hierarchical Inheritance.

ANS:

```
// Base class
class Animal {
   void eat() {
      System.out.println("Animal is eating...");
   }
}

// Derived class 1
class Dog extends Animal {
   void bark() {
      System.out.println("Dog is barking...");
   }
}

// Derived class 2
class Cat extends Animal {
   void meow() {
      System.out.println("Cat is meowing...");
   }
}

public class HierarchicalInheritance {
   public static void main(String[] args) {
      // Creating objects of derived classes
      Dog dog = new Dog();
      Cat cat = new Cat();
```

```
    // Calling methods of base class
    dog.eat();  // Dog inherits eat() from Animal
    cat.eat();  // Cat inherits eat() from Animal

    // Calling methods specific to derived classes
    dog.bark(); // Dog-specific method
    cat.meow(); // Cat-specific method
  }
}
```

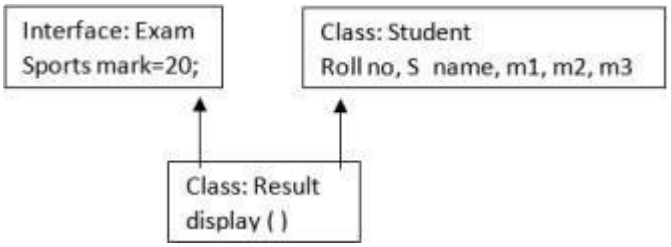
16. Develop an “Interest” interface, which contains Simple Interest and Compound Interest methods and static final field of rate 25%. Write a class to implement those methods.

ANS:

```
// Interest interface
interface Interest {
  double rate = 0.25; // Static final field representing the interest rate

  // Method to calculate Simple Interest
  double calculateSimpleInterest(double principal, double time);

  // Method to calculate Compound Interest
  double calculateCompoundInterest(double principal, double time);
}

// Class to implement the Interest interface
class InterestCalculator implements Interest {
  // Implementation of calculateSimpleInterest method
  @Override
  public double calculateSimpleInterest(double principal, double time) {
    return (principal * rate * time) / 100;
  }

  // Implementation of calculateCompoundInterest method
  @Override
  public double calculateCompoundInterest(double principal, double time) {
    return principal * (Math.pow(1 + rate / 100, time)) - principal;
  }
}

public class Demo {
  public static void main(String[] args) {
    InterestCalculator calculator = new InterestCalculator();
    double principal = 1000;
    double time = 5;

    // Calculate and display Simple Interest
    double simpleInterest = calculator.calculateSimpleInterest(principal, time);
    System.out.println("Simple Interest: " + simpleInterest);
```

```
        // Calculate and display Compound Interest
        double compoundInterest = calculator.calculateCompoundInterest(principal, time);
        System.out.println("Compound Interest: " + compoundInterest);
    }
}
```

17. Write a program to implement the following inheritance. Refer fig. no 1.



ANS:

```
interface Exam {
    int SportsMark = 20;
}

// Class Student
class Student{
    int rollNo;
    String sName;
    double m1, m2, m3;

    // Constructor
    Student(int rollNo, String sName, double m1, double m2, double m3) {
        this.rollNo = rollNo;
        this.sName = sName;
        this.m1 = m1;
        this.m2 = m2;
        this.m3 = m3;
    }

    public void display() {
        System.out.println("Student Details:");
        System.out.println("Roll No: " + rollNo);
        System.out.println("Name: " + sName);
        System.out.println("Marks: " + m1 + ", " + m2 + ", " + m3);
    }
}
// Class Result
class Result extends Student implements Exam {
    Result(int rollNo, String sName, double m1, double m2, double m3) {
        super(rollNo, sName, m1, m2, m3);
    }
```

```
    public void display() {
        super.display();
        System.out.println("SPORTS MARKS: "+SportsMark);
    }
}

public class Demo {
    public static void main(String[] args) {
        // Creating a Student object
        Result r = new Result(101, "Aditya", 100, 95, 94);

        // Displaying result using multiple inheritance
        r.display();
    }
}
```

18. Develop a program to create a class “Book” having data members “author”, “title” and “price”. Derive a class “BookInfo” having data member “stockposition” and “method to initialize and display the information for three objects.

ANS:

```
// Book class
class Book {
    // Data members
    String author;
    String title;
    double price;

    // Parameterized constructor
    Book(String author, String title, double price) {
        this.author = author;
        this.title = title;
        this.price = price;
    }
}

// BookInfo class inheriting from Book
class BookInfo extends Book {
    // Additional data members
    int stockPosition;

    // Parameterized constructor
    BookInfo(String author, String title, double price, int stockPosition) {
        super(author, title, price); // Call superclass constructor
        this.stockPosition = stockPosition;
    }

    // Method to display book information including stock position
    void displayInfo() {
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Price: " + price);
        System.out.println("Stock Position: " + stockPosition);
```

```
        System.out.println();
    }
}

public class Demo {
    public static void main(String[] args) {
        // Create three BookInfo objects
        BookInfo book1 = new BookInfo("Author1", "Title1", 20.5, 100);
        BookInfo book2 = new BookInfo("Author2", "Title2", 25.75, 150);
        BookInfo book3 = new BookInfo("Author3", "Title3", 30.0, 200);

        // Display information for each book
        System.out.println("Book 1 Information:");
        book1.displayInfo();

        System.out.println("Book 2 Information:");
        book2.displayInfo();

        System.out.println("Book 3 Information:");
        book3.displayInfo();
    }
}
```

19. Write a program, to create a class “Salary” with data members “empid”, “name” and “basicsalary”. Write an interface “Allowance” which stores rates of calculation for da 90% of basic salary, hra as 10% of basic salary and pf as 8.33% of basic salary. Include a method to calculate net salary and display it.

ANS:

```
// Interface Allowance
interface Allowance {
    double DA_RATE = 0.9;    // DA rate (90% of basic salary)
    double HRA_RATE = 0.1;   // HRA rate (10% of basic salary)
    double PF_RATE = 0.0833; // PF rate (8.33% of basic salary)
}

// Class Salary
class Salary implements Allowance {
    // Data members
    int empid;
    String name;
    double basicSalary;

    // Parameterized constructor
    Salary(int empid, String name, double basicSalary) {
        this.empid = empid;
        this.name = name;
        this.basicSalary = basicSalary;
    }
```

```
    // Method to calculate net salary
    double calculateNetSalary() {
        double da = DA_RATE * basicSalary;
        double hra = HRA_RATE * basicSalary;
        double pf = PF_RATE * basicSalary;
        double netSalary = basicSalary + da + hra - pf;
        return netSalary;
    }

    // Method to display salary details
    void displaySalary() {
        System.out.println("Employee ID: " + empid);
        System.out.println("Name: " + name);
        System.out.println("Basic Salary: " + basicSalary);
        System.out.println("DA: " + DA_RATE * basicSalary);
        System.out.println("HRA: " + HRA_RATE * basicSalary);
        System.out.println("PF: " + PF_RATE * basicSalary);
        System.out.println("Net Salary: " + calculateNetSalary());
    }
}

public class Demo {
    public static void main(String[] args) {
        // Create a Salary object
        Salary employee1 = new Salary(101, "John", 50000);

        // Display salary details
        employee1.displaySalary();
    }
}
```

20. Write a program that throws an exception called “NoMatchException” when a string is not equal to “India”.

ANS:

```
import java.util.*;

class NotMatchException extends Exception {
    NotMatchException(String msg) {
        super(msg);
    }
}

class Demo {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        try {
            String s3 = "India";
            String s;
            System.out.println("Enter the password");
            s = sc.next();
            if (s.equals(s3)) {
```

```
            System.out.println("Match done");
          } else {
            throw new NotMatchException("Invalid Match");
          }
        } catch (NotMatchException w) {
          System.out.println(w);
        } finally {
          System.out.println("THANKS FOR USING OUR STRING CHECKER ! ");
        }
      }
    }
```

21. Write a program to create a user defined exception in java.

ANS:

```
    import java.util.*;

    class MyException extends Exception {
      MyException(String msg) {
        super(msg);
      }
    }

    class Demo {
      public static void main(String args[]) throws ArithmeticException {
        Scanner sc = new Scanner(System.in);
        try {
          int a;
          System.out.println("Enter the age");
          a = sc.nextInt();
          if (a > 0) {
            System.out.println("Valid age");
          } else {
            throw new MyException("Invalid Age");
          }
        } catch (MyException w) {
          System.out.println(w);
        } finally {
          System.out.println("INSIDE FINALLY BLOCK OF EXCEPTION CODE");
        }
      }
    }
```

Output-

22. Write a program to print even and odd number using two threads with delay of 1000ms after each number?

ANS:

```
import java.util.*;

class EvenThread extends Thread{
    public void run(){
        for(int i=2;i<=50;i+=2){
            System.out.println("EVEN NUMBER: "+i);
            try {
              Thread.sleep(1000);
            } catch (Exception e) {
              System.out.println(e);
            }
        }
    }
}
class OddThread extends Thread{
   public void run(){
        for(int i=1;i<=50;i+=2){
            System.out.println("ODD NUMBER: "+i);
            try {
              Thread.sleep(1000);
            } catch (Exception e) {
              System.out.println(e);
            }
        }
   }
}

class Demo{
   public static void main(String args[]){
     EvenThread e=new EvenThread();
     OddThread o=new OddThread();
     e.start();
     o.start();
   }
}
```

23. Define an exception called “NoMatchException” that is thrown when the password accepted is not equal to “MSBTE”. Write the program.

Ans:

```
import java.util.*;

class NotMatchException extends Exception {
   NotMatchException(String msg) {
      super(msg);
```

```
    }
}

class Demo {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        try {
            String s3 = "MSBTE";
            String s;
            System.out.println("Enter the password");
            s = sc.next();
            if (s.equals(s3)) {
                System.out.println("AUTENTICATED ");
            } else {
                throw new NotMatchException("wrong password !");
            }
        } catch (NotMatchException w) {
            System.out.println(w);
        } finally {
            System.out.println("THANKS FOR USING OUR STRING CHECKER ! ");
        }
    }
}
```

24. Write a Java program in which thread A will display the even numbers between1 to 50 and thread B will display the odd numbers between1 to 50. After 3 iterations thread A should go to sleep for 50ms.

ANS:

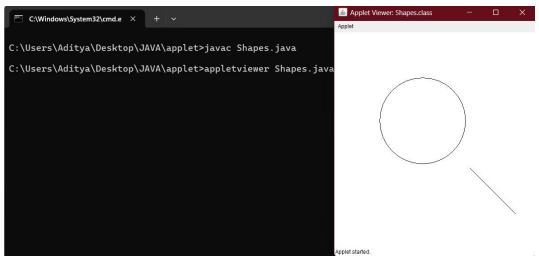
```
import java.util.*;

class EvenThread extends Thread{
    public void run(){
        for(int i=2;i<=50;i+=2){
            System.out.println("EVEN NUMBER: "+i);

                try {
                  if (i%6==0) {
                  Thread.sleep(50);
                  }
                } catch (Exception e) {
                  System.out.println(e);
                }
         }
    }
}
class OddThread extends Thread{
    public void run(){
        for(int i=1;i<=50;i+=2){
            System.out.println("ODD NUMBER: "+i);
        }
```

```
    }
}

class Demo {
    public static void main(String args[]) throws InterruptedException{
        EvenThread e=new EvenThread();
        OddThread o=new OddThread();
        e.start();
        o.start();
    }
}
```

25. Write an applet program for following graphics method. i) Drawoval () ii) Drawline ().

ANS:

```
/*<applet code=Shapes.class width=600 height=600> </applet>*/

import java.awt.*;
import java.applet.*;
public class Shapes extends Applet
{
public void paint (Graphics g)
{
   g.drawLine(400,400,300,300);
g.drawOval(100,100,190,190);
}
}
```



26. Write a java applet to display the following output in Red.

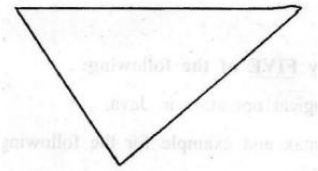
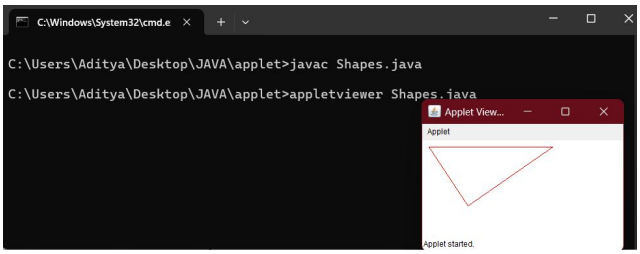


Fig No. 1.

ANS:

```
/*<applet code=Shapes.class width=600 height=600> </applet>*/

import java.awt.*;
import java.applet.*;
public class Shapes extends Applet
{
   public void paint(Graphics g)
   {
   int x[]={10,200,70};
   int y[]={10,10,100};
   g.setColor(Color.red);
   g.drawPolygon(x,y,3);
   }
   }
```



27. Write a program to read a file and then count number of words.

ANS:

```
import java.io.*;

public class Demo {
   public static void main(String are[]) throws IOException {
      int wc = 0;
      FileReader fr = new FileReader("a.txt");
      int c = 0;
      try {
         while (c != -1) {
            if ((c=fr.read())!=-1)
               wc++;
         }
         System.out.println("Number of words :" + wc);
      } finally {
         if (fr != null)
            fr.close();
      }
   }
```

```
}
```

28. Write a program to append content of one file into another file.

ANS:

```
import java.io.*;

public class Demo {
   public static void main(String are[]) throws IOException {
      FileWriter fw = new FileWriter("a.txt",true);
      String str="world";
      try {
         fw.write(str);
         System.out.println("data appended to file successfully");
         fw.close();
      } finally {
         if (fw != null)
            fw.close();
      }
   }
}
```

29. Write a program for reading and writing character to and from the given files using character stream classes.

ANS:

```
import java.io.*;

public class Demo {
   public static void main(String[] args) {
      // Define the file names
      String inputFile = "a.txt";
      String outputFile = "output.txt";

      // Write characters to a file
      try (FileWriter writer = new FileWriter(inputFile)) {
         writer.write("Hello, this is a test.\n");
         writer.write("Writing characters to a file using FileWriter.\n");
         writer.write("This is the third line.\n");
      } catch (IOException e) {
         System.err.println("Error writing to file: " + e.getMessage());
      }

      // Read characters from the file and write them to another file
      try (FileReader reader = new FileReader(inputFile);
          FileWriter writer = new FileWriter(outputFile)) {
         int character;
         while ((character = reader.read()) != -1) {
            writer.write(character);
         }
      } catch (IOException e) {
         System.err.println("Error reading/writing file: " + e.getMessage());
      }
```

```
        // Display the contents of the output file
        try (BufferedReader br = new BufferedReader(new FileReader(outputFile))) {
            System.out.println("Contents of the output file:");
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.err.println("Error reading output file: " + e.getMessage());
        }
    }
}
```

30. Write a Java program to copy the content of one file into another.

ANS:

```
import java.io.*;

class Demo {
    public static void main(String args[]) throws IOException {
        FileReader fr = new FileReader("a.txt");
        FileWriter fo = new FileWriter("output.txt");
        int ch;
        try {
            while ((ch = fr.read()) != -1) {
                fo.write(ch);
            }
            System.out.println("file copied successfully");
            fr.close();
            fo.close();
        } finally {
            if (fr != null)
                fr.close();
            if (fo != null)
                fo.close();
        }
    }
}
```

31. Write a Java program to count the number of words from a text file using stream classes.

ANS:

```
import java.io.*;

public class Demo {
    public static void main(String are[]) throws IOException {
```

```
        int wc = 0;
        FileReader fr = new FileReader("a.txt");
        int c = 0;
        try {
           while (c != -1) {
              if ((c=fr.read())!=-1)
                 wc++;
           }
           System.out.println("Number of words :" + wc);
        } finally {
           if (fr != null)
              fr.close();
        }
     }
}
```