**1.Write programs using the primitive data types.**

class GFG {

       public static void main(String args[])

     {

         char a = 'G';

int i = 89;

         byte b = 4;

         short s = 56;

         double d = 4.355453532;

         float f = 4.7333434f;

         long l = 12121;

         System.out.println("char: " + a);

         System.out.println("integer: " + i);

         System.out.println("byte: " + b);

         System.out.println("short: " + s);

         System.out.println("float: " + f);

         System.out.println("double: " + d);

         System.out.println("long: " + l);

    }

}

**2.a. Java Program to print biggest element among three integers**

import java.util.\*;

class Biggest

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter any three integers:");

int a=sc.nextInt();

int b=sc.nextInt();

int c=sc.nextInt();

if(a>b && a>c)

System.out.print(" a is big");

else

if(b>c)

System.out.print("b is big");

else

System.out.print("c is big");

}

}

**2.b. Java program to print factorial value of the given integer**

import java.util.\*;

class Factorial

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter n value:");

int n=sc.nextInt();

int f=1;

for(int i=1;i<=n;i++)

{

f=f\*i;

}

System.out.print("Factorial value of the given integer:"+f);

}

}

**2.c. java program to print sum of the following series**

import java.util.\*;

class Series

{

public static void main(String s[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter x,n values:");

int x=sc.nextInt();

int n=sc.nextInt();

int i=0;

double sum=0;;

while(i<=n)

{

sum=sum+Math.pow(x,i);

i++;

}

System.out.print("Sum is :"+sum);

}

}

**2.d. Java program to print Triangle**

import java.util.\*;

class Triangle

{

public static void main(String[] args)

{

int i,j;

for(i=1;i<=5;i++)

{

for(j=1;j<=i;j++)

System.out.print(j);

System.out.print("\n");

}

}

}

**3.a.Write a program to give values to variable interactively keyboard**

**(Use readLine() method)**

import java.io.\*;

public class read

{

public static void main(String[] args)

{

Console console=System.console();

if(console==null)

{

System.out.println("console is not available to current jvm");

return;

}

String username=console.readLine("enetr username");

System.out.println("entered username"+username);

}

}

**3.b.Write a program to give values to variable interactively keyboard**

**(Use readLine() method for password)**

import java.io.\*;

public class GFG

{

    public static void main(String[] args)

    {

        Console cnsl= System.console();

          if (cnsl == null) {

            System.out.println( "No console available");

            return;

        }

        String str = cnsl.readLine("Enter username : ");

         System.out.println( "Username : " + str);

        char[] ch = cnsl.readPassword( "Enter password : ");

         System.out.println( "Password : " + ch);

    }

}

**3.c. Java program Reading and Writing Primitive data types using DataInputStream and DataOoutputStream**

import java.io.\*;

class DataInputStreamDemo

{

    public static void main(String args[]) throws IOException

{

try ( DataOutputStream dout = new DataOutputStream(new FileOutputStream("file.dat")) )

{

            dout.writeDouble(1.1);

            dout.writeInt(55);

            dout.writeBoolean(true);

            dout.writeChar('4');

         }

         catch (FileNotFoundException ex)

{

            System.out.println("Cannot Open the Output File");

            return;

         }

        try ( DataInputStream din =new DataInputStream(new FileInputStream("file.dat"))

{

            double a = din.readDouble();

            int b = din.readInt();

            boolean c = din.readBoolean();

            char d = din.readChar();

           System.out.println("Values: " + a + " " + b + " " + c + " " + d);

         }

          catch (FileNotFoundException e)

{

System.out.println("Cannot Open the Input File");

return;

         }  }}

**3.d.Write a Java Program for Reading and Writing operations on files using File streams**

**Writing a File:**

import java.io.\*;

import java.io.File.\*;

class writetofile

{

public static void main(String args[])

{

try

{

fileWriter fwrite=new fileWriter(“d:fileoperationexample.txt”);

fwrite.write(“a named location used to stote related information the file “);

fwrite.close();

System.out.println(“content successful”);

}

catch(IOException e)

{

System.out.println(“unexpected error”);

e.printstacktrace();

}

}

**}**

**Reading a File :**

import java.io.\*;

import java.io.\*;

import java.util.scanner;

class readfromfile

{

public static void main(String args[])

{

try

{

file f1=new file(” d:fileoperationexample.txt”);

Scanner datareader=new Scanner(f1);

while(datareader.hasnextLine())

{

String filedata=datareader.nextLine();

System.out.println(filedata);

}

datareader.close();

}

catch(FileNotFoundException exception)

{

System.out.println(“unexpected error”);

exception.printstackTrace();

}}**}**

**4. Java program to perform string manipulation using String class**

class StringLib

{

public static void main(String s[])

{

String s1=new String("Vikas");

String s2=new String("College");

String s3=new String();

System.out.println("\n Length of the first string:"+s1.length());

System.out.println("Length of the second string:"+s2.length());

System.out.println("Is (s1==s2):"+s1.equals(s2));

s3=s1.concat(s2);

System.out.println("Concatenation :"+s3);

System.out.println("Upper case :"+s3.toUpperCase());

System.out.println("Character at 2 in s3:"+s3.charAt(2));

System.out.println("Index of c in s3: "+s3.indexOf('C'));

System.out.println("Sub string: "+s3.substring(3,9));

}

}

**5.a.Write a java program to print multiplication of the given two matrices(Create a class and objects)**

import java.util.\*;

class Matrix

{

int a[][],b[][],c[][];

int i,j,k,r1,c1,r2,c2;

void getMatrix()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter order of the first matrix:");

r1=sc.nextInt();

c1=sc.nextInt();

System.out.print("Enter order of the second matrix:");

r2=sc.nextInt();

c2=sc.nextInt();

if(c1==r2)

{

a=new int[r1][c1];

b=new int [r2][c2];

System.out.print("Enter elements of first matrix:");

for(i=0;i<r1;i++)

for(j=0;j<c1;j++)

a[i][j]=sc.nextInt();

System.out.print("Enter elements of second matrix:");

for(i=0;i<r2;i++)

for(j=0;j<c2;j++)

b[i][j]=sc.nextInt();

multiplication();

putMatrix();

}

else

System.out.print("Matrix addition is not possible");

}

void multiplication()

{

c=new int [r1][c2];

for(i=0;i<r1;i++)

for(j=0;j<c2;j++)

{

for(k=0;k<c1;k++)

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

}

void putMatrix()

{

System.out.println("Multiplicaiton is :");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

System.out.print(c[i][j]+"\t");

System.out.print("\n");

}

}

}

class mat

{

public static void main(String s[])

{

Matrix m1=new Matrix();

m1.getMatrix();

}

}

**5.b. Java program read Employ details and then increment employ salary then print employ details (Create a class and objects)**

import java.util.\*;

class Employ

{

int eid;

String ename;

int salary;

void getEmploy(int id,String name,int sal)

{

eid=id;

ename=name;

salary=sal;

}

void increment(int amount)

{

salary=salary+amount;

}

void putEmploy()

{

System.out.print("\nEmploy ID:"+eid);

System.out.print("\nEmploy Name:"+ename);

System.out.print("\nEmploy Salary:"+salary);

}

}

class Emp

{

public static void main(String s[])

{

Employ e1=new Employ();

e1.getEmploy(1,"john", 4000);

e1.increment(2050);

e1.putEmploy();

}

}

**6.a)Write a Java Program using Default constructor:**

Class Bike1

{

Bike1()

{

System.out.println( “Bike is created”);

}

Public static void main(String args[])

{

Bike1 b=new Bike1();

}

}

**6.b)Write a Java program for Parameterized constructor:**

class Student

{

int id;

String name;

Student(int i, String n)

{

id=i;

name=n;

}

void display()

{

System.out.println(id +" " +name);

}

public static void main(String args[])

{

Student s1=new Student(111,"kiran");

Student s2=new Student(222,"aryan");

s1.display();

s2.display();

}}

**6.c)Write a Java program for Constructor Overloading:**

class Student

{

int id;

String name;

int age;

Student(int i,String n)

{

id=i;

name=n;

}

Student(int i,String n,int a)

{

id=i;

name=n;

age=a;

}

void display()

{

System.out.println(id+""+name+""+age);

}

public static void main(String args[])

{

Student s1=new Student(111,"kiran");

Student s2=new Student(222,"arya",25);

s1.display();

s2.display();

}

}

**7.Write a Java program for using Command line arguments :**

class c

{

public static void main ( String args[])

{

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

System .out.println(args[i]);

}

}

**8.a.Write a java Program for Method overloading** :

public class MethodOverloadingExample

{

public static void main(String[] args)

{

MethodOverloadingExample example = new MethodOverloadingExample(); example.display();

example.display(10);

example.display("Hello");

}

public void display()

{

System.out.println("Display method with no parameters");

}

public void display(int num)

{

System.out.println("Display method with int parameter: " + num);

}

public void display(String text)

{

System.out.println("Display method with String parameter: " + text);

}

}

**8.b. java program to print sum of the integer array and sum of the float array using method overloading**

import java.util.\*;

class ArraySum

{

static int sum(int a[],int size)

{

int s=0;

for(int i=0;i<size;i++)

s=s+a[i];

return s;

}

static float sum(float b[],int size)

{

float s=0;

for(int i=0;i<size;i++)

ijs=s+b[i];

return s;

}

public static void main(String s[])

{

int x[]={4,5,1,2,9};

float y[]={2.1f, 5.2f, 3.4f};

int p=sum(x,5);

float q=sum(y,3);

System.out.print("\nSum of the integer array elements :" +p);

System.out.print("\nSum of the float array elements : "+q);

}

}

**9.a. java program for student and course details using single inheritance**

import java.util.\*;

class Course

{

int code;

String cname;

String branch;

void getCourse()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter Course code, name , branch=");

code=sc.nextInt();

cname=sc.next();

branch=sc.next();

}

}

class Student extends Course

{

int sno;

String sname;

void getStudent()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter sno, name:");

sno=sc.nextInt();

sname=sc.next();

}

void putStudent()

{

System.out.print("\n Student Number:"+sno);

System.out.print("\n Student Name:"+sname);

System.out.print("\n Course Code:"+code);

System.out.print("\n Course Name:"+cname);

}

}

class stu

{

public static void main(String s[])

{

Student s1=new Student();

s1.getStudent();

s1.getCourse();

s1.putStudent();

}

}

**9.b. Java program for student, course and marks details using multilevel inheritance**

import java.util.\*;

class Course

{

int code;

String cname;

String branch;

void getCourse()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter Course code, name , branch=");

code=sc.nextInt();

cname=sc.next();

branch=sc.next();

}

}

class Marks extends Course

{

int sub1,sub2,sub3,total;

float avg;

void getMarks()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter three subjects marks:");

sub1=sc.nextInt();

sub2=sc.nextInt();

sub3=sc.nextInt();

}

void calc()

{

total=sub1+sub2+sub3;

avg=(float)total/3;

}

}

class Student extends Marks

{

int sno;

String sname;

void getStudent()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter sno, name:");

sno=sc.nextInt();

sname=sc.next();

}

void putStudent()

{

System.out.print("\n Student Number:"+sno);

System.out.print("\n Student Name:"+sname);

System.out.print("\n Course Code:"+code);

System.out.print("\n Course Name:"+cname);

System.out.print("\n Total marks:"+total);

System.out.print("\n Average :"+avg);

}

}

class stu

{

public static void main(String s[])

{

Student s1=new Student();

s1.getStudent();

s1.getCourse();

s1.getMarks();

s1.calc();

s1.putStudent();

}

}

**10.a.Java Program for Method overriding**

class A

{

void add(int a, int b)

{

System.out.println(“Sum of two numbers in super" + (a+b));

}

}

class B extends A

{

void add( int a, int b)//override

{

System.out.println("Sum of two numbers in subclass"+(a+b));

}

}

public class c

{

public static void main(String args[])

{

A a1=new A();

a1.add(10,20);

B b1=new B();

b1.add(30,20);

}}

**10.b.Write a Java program for use of super keyword(variable):**

class base

{

int num=30;

}

class derived extends base

{

int num=20;

void callthis()

{

System.out.println("base num"+super.num);

System.out.println("derived num"+num);

}

}

class c

{

public static void main(String args[])

{

derived temp=new derived();

temp.callthis();

}

}

**10.c.Write a Java program for use of super keyword(method):**

class base

{

void display()

{

System.out.println("base class");

}

}

class derived extends base

{

void callthis()

{

super.display();

display();

}

void display()

{

System.out.println("derived class");

}

}

class c

{

public static void main(String args[])

{

derived temp=new derived();

temp.callthis();

}

}

**10.d.Write a Java program for use of this keyword**

class person

{

int id;

String name;

person(int id,String name)

{

this.id=id;

this.name=name;

}

}

class emp extends person

{

float salary;

emp(int id,String name,float salary)

{

super(id,name);

this.salary=salary;

}

void display()

{

System.out.println(id + ""+name + ""+salary);

}

}

class c

{

public static void main(String args[])

{

emp e1=new emp(1,"ankit",45000);

e1.display();

}

}

**10.e.Write a java program for Method overriding**

class Vehicle

{

public void start()

{

System.out.println("Vehicle started");

}

}

class Car extends Vehicle

{

public void start()

{

System.out.println("Car started");

}

}

public class MethodOverridingExample

{

public static void main(String[] args)

{

Vehicle vehicle = new Vehicle();

vehicle.start();

Car car = new Car();

car.start();

}}

**11.a.Write a Java Program for Create a Package(using Package Name):**

**//Save by a.java**

package pack;

public class A

{

Public void msg()

{

System.out.println(“hello”);

}

}

**//Save by b.java**

Package mypack;

import pack.\*;

Class B

{

public static void main(String args[])

{

A obj=new A();

obj.msg();

}

}

**11.b.Write a Java Program for Create a Package(Using packagename.classname):**

**Create a Package:**

**//Save by a.java**

package pack;

public class A

{

Public void msg()

{

System.out.println(“hello”);

}

}

**//Save by b.java**

Package mypack;

import pack.A;

Class B

{

public static void main(String args[])

{

A obj=new A();

obj.msg();

}

}

**11.cWrite a java program for List off Access specifier:**

1. **Private**

class simple

{

private int data=40;

private void msg()

{

System.out.println("hello");

}

}

class c

{

public static void main(String args[])

{

simple obj=new simple();

System.out.println(obj.data);

obj.msg();

//}

}

**Public modifier:**

**//Saved by a.java**

package pack;

public class a

{

public void msg()

{

System.out.println(“hello”);

}

}

**//Saved b.java**

import pack.\*;

class b

{

public static void main(String args[])

{

a obj=new a();

obj.msg();

}

}

**Protected :**

**Saved as A.java**

Package pack;

Public class A

{

Protected void msg()

{

System.out.println(“hello”);

}

}

**Saved as B.java**

import pack.\*;

Class B extends A

{

Public static void main(String args[])

{

B obj=new B();

obj.msg();

}

}

**11.d. java program to create a Package and importing a package**

**Even.java**

package series;

public class Even

{

void evenSeries(int n)

{

for(int i=0;i<=n;i++)

if(i%2==0)

System.out.print(i+"\t");

}

}

**Importing created package**

**EvenNumbers.java**

import series.Even;

import java.util.\*;

class EvenNumbers

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter n value:");

int n=sc.nextInt();

Even.evenSeries(n); }}

**12.Write a Java program on Interfaces(Multiple inheritance)**

interface A

{

void displayA();

}

interface B

{

void displayB();

}

class AB implements A,B

{

public void displayA()

{

System.out.println(“a class”);

}

public void displayB()

{

System.out.println(“b class”);

}

}

class multiple

{

public static void main(string args[])

{

AB a1=new AB();

a1.displayA();

a1.displayB();

}

}

**13.aWrite a Java Program for to search a student mark percentage based on pin number using Array List .**

import java.util.ArrayList;

import java.util.Scanner;

class Student

{

private int pinNumber;

private double marks;

public Student(int pinNumber, double marks)

{

this.pinNumber = pinNumber;

this.marks = marks;

}

public int getPinNumber()

{

return pinNumber;

}

public double getMarks()

{

return marks;

}

}

public class StudentSearchProgram

{

public static void main(String[] args)

{

ArrayList<Student> students = new ArrayList<>();

students.add(new Student(123, 85.5));

students.add(new Student(456, 76.8));

students.add(new Student(789, 92.3));

students.add(new Student(101, 68.9));

double percentage=0.0;

int flag=0;

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the PIN number to search: ");

int searchPin = scanner.nextInt();

for (Student student : students)

{

if (student.getPinNumber() == searchPin)

{

percentage=(student.getMarks() / 100) \* 100;

flag=1;

}

}

if (flag==1)

{

System.out.println("Percentage: " + percentage + "%");

}

else

{

System.out.println("Student with PIN number " + searchPin + " not found.");

}

}

}

**13.b.Write a Java program for to search an element from hash table**

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

class HashTableSearchProgram

{

public static void main(String[] args)

{

HashMap<Integer, String> hashTable = new HashMap<>();

hashTable.put(1, "Apple");

hashTable.put(2, "Banana");

hashTable.put(3, "Orange");

hashTable.put(4, "Mango");

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the key to search: ");

int searchKey = scanner.nextInt();

if (hashTable.containsKey(searchKey))

{

System.out.println("Element found: " + hashTable.get(searchKey));

}

else

{

System.out.println("Element not found.");

}

}

}

**13.c.Write a Java Program for sorting employ details using hash map**

import java.util.\*;

class sortmapKey

{

static Map<String, Integer> map = new HashMap<>();

public static void main(String args[])

{

map.put("Jayant", 80);

map.put("Abhishek", 90);

map.put("Anushka", 80);

map.put("Amit", 75);

map.put("Danish", 40);

TreeMap<String, Integer> sorted = new TreeMap<>();

sorted.putAll(map);

for (Map.Entry<String, Integer> entry : sorted.entrySet())

System.out.println("Key = " + entry.getKey() +", Value = " +entry.getValue());

}

}

**14.a.Write a java program for Exception handling(using try and catch).**

public class testcatch

{

public static void main(String args[])

{

try

{

int data=50/0;

}

catch(ArithmeticException e)

{ {

System.out.println(e);

}

System.out.println("rest of the code......");

}

}

**14.b.Write a Java program for use exception handling(using try,catch and finally)**

class testfinal

{

public static void main(String args[])

{

try

{

int data=25/5;

System.out.println(data);

}

catch(NullPointerException e)

{

System.out.println(e);

}

finally

{

System.out.println("finally block is always executed");

}

System.out.println("rest of the code........");

}

}

**14.c .Write a Java program for multiple catch statements in java**

class multi\_catch

{

public static void main(String args[])

{

int arr[]={5,10};

int b=5;

try

{

int x=arr[2]/(arr[0]-b);

}

catch(ArithmeticException ae)

{

System.out.println("divide by zero is not possible");

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("can not possible to access an element out of array bounds");

}

catch(ArrayStoreException e)

{

System.out.println("can not store different type of values");

}

finally

{

int y=arr[1]/arr[0];

System.out.println("y="+y);

}

}

}

**14.d . Write a java program for nested try in exception handling**

class nestedtry

{

public static void main(String args[])

{

try

{

int arr[]={1,0,4,2};

try

{

int x=arr[3]/arr[1];

}

catch(ArithmeticException ae)

{

System.out.println("divide by zero not possible");

}

arr[4]=3;

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("array index out of bounds exception");

}

}

}

**15.a.Write a java program for create a Thread (Using Runnable interface)**

class t1 implements Runnable

{

public void run()

{

System.out.println("thread is running");

}

public static void main(String args[])

{

t1 obj1=new t1();

Thread t=new Thread(obj1);

t.start();

}

}

**15.b. Write a java program for create a Thread (Using Thread class)**

class t2 extends Thread

{

public void run()

{

System.out.println("thread is running");

}

public static void main(String args[])

{

t2 obj1=new t2();

obj1.start();

}

}

**15.c.Write a java program for multithreading**

class t1 extends Thread

{

public void run()

{

int i;

System.out.println("even numbers");

for(i=0;i<=10;i++)

{

if(i%2==0)

{

System.out.println(i);

}

}

}

}

class t2 extends Thread

{

public void show()

{

int i;

System.out.println("odd numbers");

for(i=0;i<=10;i++)

{

if(i%2!=0)

{

System.out.println(i);

}

}

}

}

class t4

{

public static void main(String args[])

{

t1 obj1=new t1();

t2 obj2=new t2();

obj2.show();

obj1.run();

}

}

**15.d.Write a java program for Inter Thread Communication**

class customer

{

int amount=0;

int flag=0;

synchronized int withdraw(int r)

{

if(flag==0)

{

try

{

System.out.println(“waiting….”);

wait();

}

catch(exception e)

{}

}

amount=amount-r;

System.out.println(“withdraw complete”);

return amount;

}

synchronized void deposit(int r)

{

System.out.println(Thread.currentThread().getName()+”: is going to deposit”);

amount=amount+r;

notifyAll();

System.out.println(“deposit complete”);

flag=1;

}

}

public class inter

{

public static void main(string args[])

{

final customer c=new customer();

Thread t1=new thread()

{

public void run()

{

c.withdraw(5000);

System.out.println(“after with draw”+c.amount);

}

};

Thread t2=new thread()

{

public void run()

{

c.deposit(10000);

System.out.println(“after deposit”+c.amount);

}

};

t1.start();

t2.start();

}

}

**16.a.Write a java program for Graphics Drawing a Line(Using Applet)**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code=”lines” width=300 height=200>

</applet>

\*/

public class lines extends Applet

{

public void init()

{

setBackground(Color.black);

setForeground(Color.green);

}

public void paint(Graphics g)

{

g.drawLine(0,0,100,100);

g.drawLine(0,100,100,0);

g.drawLine(40,25,250,180);

g.drawLine(75,l90,400,400);

g.drawLine(20,150,400,400);

g.drawLine(5,290,80,19);

}}

**16.b.Write a java program for Graphics Drawing a Rectangle(Using Applet)**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code=”rectangles” width=300 height=200>

</applet>

\*/

public class rectangles extends Applet

{

public void init()

{

setBackground(Color.black);

setForeground(Color.green);

}

public void paint(Graphics g)

{

g.drawRect(10,10,60,50);

g.fillRect(100,10,60,50);

g.drawRoundRect(190,10,60,50,15,15);

g.fillRoundRect(70,90,140,100,30,40);

}

}

**16.c. Write a java program for Graphics Drawing a ellipse(Using Applet)**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code=”ovals” width=300 height=200>

</applet>

\*/

public class ovals extends Applet

{

public void init()

{

setBackground(Color.black);

setForeground(Color.green);

}

public void paint(Graphics g)

{

g.drawOval(10,10,50,50);

g.fillOval(100,10,75,50);

g.drawOval(190,10,90,30);

g.fillOval(70,90,140,100);

}

}

**16.d. Write a java program for Graphics Drawing a Polygons(Using Applet)**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code=”Polygons” width=230 height=210>

</applet>

\*/

public class Polygons extends Applet

{

public void init()

{

setBackground(Color.black);

setForeground(Color.green);

}

public void paint(Graphics g)

{

int[][] xPoints={{50,25,25,75,75},{50,25,25,75,75},{100,100,150,100,150,150,125,100,150},{100,10 0,150,100,150,150,125,100,150}};

int[][] yPoints={{10,35,85,85,35,10},{110,135,185,185,135},{85,35,35,85,85,35,10,35,85},{185,135,135,185,185,135,110,135,185}};

int[] nPoints={5,5,9,9};

g.drawPolygon(xPoints[0],yPoints[0],nPoints[0]);

g.fillPolygon(xPoints[1],yPoints[1],nPoints[1]);

}

}

**16.e.Write a java program for Colors**

import java.awt.Graphics;

import java.applet.\*;

import java.awt.Color;

/\*

<applet code=”Colors” width=200 height=200>

</applet>

\*/

public class Colors extends Applet

{

public void paint(Graphics g)

{

setBackground(Color.RED);

}

}

**16.f. Write a Java Program for animation using applet**:

**import** java.awt.\*;

**import** java.applet.\*;

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

</applet>

\*/

**public** **class** AnimationExample **extends** Applet

 {

 Image picture;

**public** **void** init()

{

    picture =getImage(getDocumentBase(),"bike1.jpg");

  }

**public** **void** paint(Graphics g)

 {

**for**(**int** i=0;i<500;i++)

{

      g.drawImage(picture, i,30, **this**);

**try**

{Thread.sleep(100);

}

**catch**(Exception e){}

    }

  }

}

**17.a.Write a java program for keyboard events using applet**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="keyevents.class" width=400 height=200>

</applet>

\*/

public class keyevents extends Applet implements KeyListener

{

String msg = " ";

public void init()

{

addKeyListener(this);

}

public void keyPressed(KeyEvent k)

{

showStatus("Key Pressed");

}

public void keyReleased(KeyEvent k)

{

showStatus("Key Released");

}

public void keyTyped(KeyEvent k)

{

msg += k.getKeyChar();

repaint();

}

public void paint(Graphics g)

{

g.drawString(msg, 20, 40);

}

}

**17.b.Write a java program for Mouse events**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

<applet code="MouseEvents" width=300 height=100>

</applet>

\*/

public class MouseEvents extends Applet

implements MouseListener, MouseMotionListener

{

String msg = "";

int mouseX = 0, mouseY = 0;

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

public void mouseClicked(MouseEvent me)

{

mouseX = 0;

mouseY = 0;

msg = "Mouse Clicked";

repaint();

}

public void mousePressed(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "Down";

repaint();

}

public void mouseReleased(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "Up";

repaint();

}

public void mouseDragged(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "\*";

showStatus("Dragging mouse at " + mouseX + "," + mouseY);

repaint();

}

public void mouseEntered(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse Entered";

repaint();

}

public void mouseExited(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse Exited";

repaint();

}

public void mouseMoved(MouseEvent me)

{

msg="mouse moved";

showStatus("Moving mouse at " + me.getX() + "," + me.getY());

}

public void paint(Graphics g)

{

g.drawString(msg, mouseX, mouseY);

}}

**17.c. Write a Java Program illustrate Text field**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*

<applet code="twonumbers" width="400" height="200">

</applet>

\*/

public class twonumbers extends Applet implements ActionListener

{

TextField firstNum,secondNum,resultNum;

public twonumbers()

{

setLayout(new GridLayout(3,2,10,15));

setBackground(Color.cyan);

firstNum=new TextField(15);

secondNum=new TextField(15);

resultNum=new TextField(15);

secondNum.addActionListener(this);

add(new Label("Enter First Number"));

add(firstNum);

add(new Label("Enter Second Numer"));

add(secondNum);

add(new Label("SUM"));

add(resultNum);

}

public void actionPerformed(ActionEvent e)

{

String str1=firstNum.getText();

double fn=Double.parseDouble(str1);

double sn=Double.parseDouble(secondNum.getText());

resultNum.setText("sum is"+(fn+sn));

}

}

**17.d.Write a Java Program illustrate Button Control**

import java.applet.Applet;

import java.awt.Button;

import java.awt.Graphics;

import java.awt.event.\*;

/\*

<applet code="HandleActionEventExample" width=300 height=200>

</applet>

\*/

public class HandleActionEventExample extends Applet implements ActionListener

{

String actionMessage="";

public void init()

{

Button Button1=new Button("Ok");

Button Button2=new Button("Cancel");

add(Button1);

add(Button2);

Button1.addActionListener(this);

Button2.addActionListener(this);

}

public void paint(Graphics g)

{

g.drawString(actionMessage,10,50);

}

public void actionPerformed(ActionEvent ae)

{

String action=ae.getActionCommand();

    if(action.equals("Ok"))

    actionMessage="Ok Button pressed";

    else if(action.equals("Cancel"))

    actionMessage="Cancel Button Pressed";

repaint();

}

}

**17.e.Write a Java Program illustrate check Box**

import java.applet.Applet;

import java.awt.Checkbox;

import java.awt.Graphics;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

/\*

<applet code="CheckboxEvent" width=200 height=200>

</applet>

\*/

public class CheckboxEvent extends Applet implements ItemListener

{

Checkbox java=null;

Checkbox vb=null;

Checkbox c=null;

public void init()

{

java=new Checkbox("Java");

vb=new Checkbox("Visual Basic");

c=new Checkbox("c");

add(java);

add(vb);

add(c);

java.addItemListener(this);

vb.addItemListener(this);

c.addItemListener(this);

}

public void paint(Graphics g)

{

g.drawString("Java:"+java.getState(),10,80);

g.drawString("VB:"+vb.getState(),10,100);

g.drawString("C:"+c.getState(),10,120);

}

public void itemStateChanged(ItemEvent ie)

{

repaint();

}

}

**17.f. Write a Java Program illustrate List Control**

import java.applet.Applet;

import java.awt.Graphics;

import java.awt.List;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

/\*

<applet code="GetSelectedItem" width=200 height=200>

</applet>

\*/

public class GetSelectedItem extends Applet implements ItemListener

{

List list=null;

public void init()

{

list=new List(5,true);

list.add("one");

list.add("two");

list.add("three");

list.add("four");

list.add("five");

list.add("six");

add(list);

list.addItemListener(this);

}

public void paint(Graphics g)

{

String[] Items=list.getSelectedItems();

String msg="";

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

{

msg=Items[i]+""+msg;

}

g.drawString("Selected Items:"+ msg, 10, 120);

}

public void itemStateChanged(ItemEvent ie)

{

repaint();

}

}