**Coding: (Prime Numbers)**

#include<iostream>

using namespace std;

int main()

{

int low,high,i,flag;

cout<<"Enter two numbers(intervals:";

cin>>low>>high;

cout<<"Prime numbers between "<<low<<" and "<<high<<" are: ";

while(low<high)

{

flag=0;

for(i=2;i<=low/2;++i){

if(low%i==0)

{

flag=1;

break;

}

}

if(flag==0)

cout<<low<<" ";

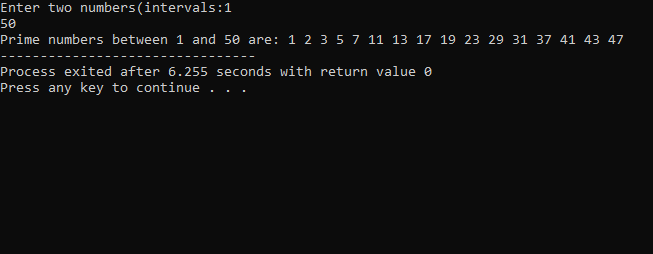
++low;

}

return 0;

}

**Output:**



**Coding: (String Operations)**

#include<iostream>

using namespace std;

#include<conio.h>

#include<string.h>

#include<ctype.h>

class STRING

{

char arr[30];

public:

void getdata();

void putdata();

STRING operator +(STRING);

int operator==(STRING);

void length();

};

void STRING::getdata()

{

cout<<"\n\n Entry any String:";

cin>>arr;

}

void STRING::putdata()

{

cout<<arr;

}

STRING STRING::operator +(STRING s)

{

STRING temp;

strcpy(temp.arr,arr);

strcat(temp.arr,s.arr);

return temp;

}

int STRING::operator ==(STRING s)

{

if(stricmp(arr,s.arr)==0)

return 1;

else

return 0;

}

void STRING::length()

{

cout<<strlen(arr);

}

int main()

{

STRING s1,s2,s3;

s1.getdata();

s2.getdata();

cout<<"\n\n First String is :";

s1.putdata();

cout<<"\n\n Second String is :";

s2.putdata();

cout<<"\n\n\n\t\t The Concatenated String";

cout<<"\n\n\n String is:";

s3=s1+s2;

s3.putdata();

cout<<"\n\n\n\t\t The Length Of Strings";

cout<<"\n\n\n First String Length:";

s1.length();

cout<<"\n\n\n Second String Length:";

s3.length();

cout<<"\n-------------------";

getch();

}

**Output:**



**Coding: (Friend Function)**

#include<iostream>

using namespace std;

#include<conio.h>

class second;

class first {

int x;

float y;

public:

void getdata()

{

cout<<"\n\n Enter the Integer Value:";

cin>>x;

cout<<"\n\n Enter the Float Value:";

cin>>y;

}

friend void show(first,second);

};

class second

{

int x;

float y;

public:

void getdata()

{

cout<<"\n\n Enter the Integer Value:";

cin>>x;

cout<<"\n\n Enter the Float value :";

cin>>y;

}

friend void show(first,second);

};

void show(first f,second s) {

cout<<"\n\n\n\n Addition of integer Value is:"<<f.x+s.x;

cout<<"\n\n Addition of Float Value is:"<<f.y+s.y;

}

int main() {

cout<<"\n------------------";

cout<<"\n\n friend function:";

cout<<"\n------------------";

first f1;

f1.getdata();

second s1;

s1.getdata();

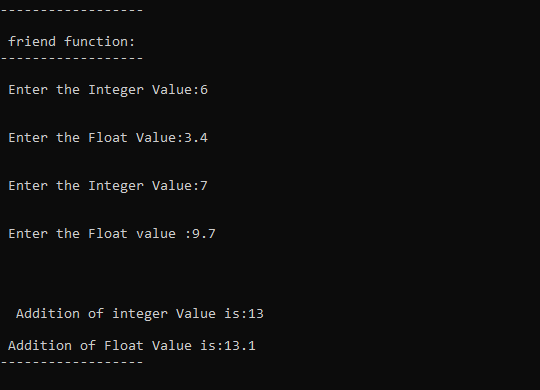
show(f1,s1);

cout<<"\n------------------";

getch();

}

**Output:**



**Coding: (Complex Numbers)**

#include<iostream>

using namespace std;

#include<conio.h>

class complex

{

private:

int real,imag;

public:

void getdata()

{

cout<<"\n\n\n Enter Real Number :";

cin>>real;

cout<<"\n\n\n Enter Imaginary Number :";

cin>>imag;

}

void display()

{

if(imag<0)

{

cout<<real<<"-"<<imag\*(-1)<<"i";

}

else

{

cout<<real<<"+"<<imag<<"i";

}

}

complex operator +(complex);

complex operator -(complex);

};

complex complex::operator +(complex obj)

{

complex temp;

temp.real=real+obj.real;

temp.imag=imag+obj.imag;

return temp;

}

complex complex::operator -(complex obj)

{

complex temp;

temp.real=real-obj.real;

temp.imag=imag-obj.imag;

return temp;

}

int main()

{

complex obj1,obj2,obj3;

cout<<"\n-------------------------";

cout<<"\t Add Two Complex Numbers by Using Opertaor Overload";

cout<<"\n\n\t\t Enter Input for object 1.";

obj1.getdata();

cout<<"\n\n\t\t Enter Input For Object 2.";

obj2.getdata();

cout<<"\n\n\n The First Complex Number :";

obj1.display();

cout<<"\n\n The Second COmplex Number :";

obj2.display();

cout<<"\n\n\n The Addition Of Two Complex Number :";

obj3=obj1+obj2;

obj3.display();

cout<<"\n\n\n The Subtraction Of Two Complex Number :";

obj3=obj1-obj2;

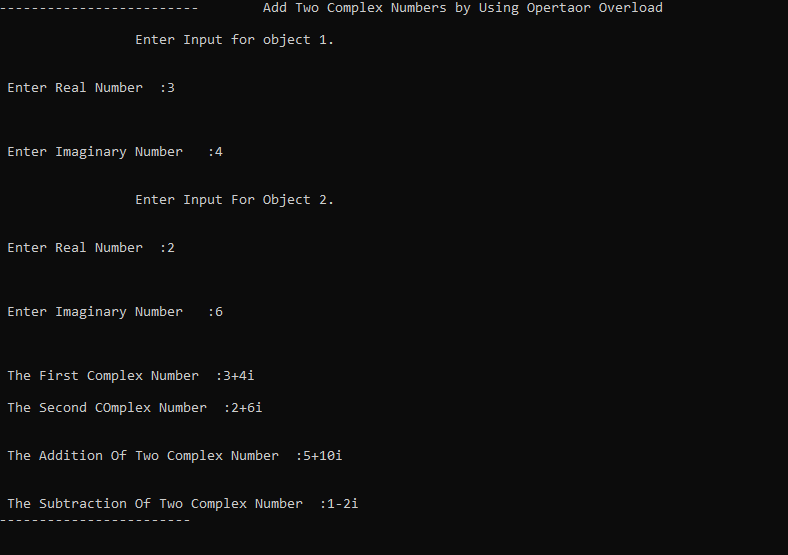
obj3.display();

cout<<"\n------------------------";

getch();

}

**Output:**



**Coding: (Employee PayRoll System using Inheritance)**

#include<iostream>

using namespace std;

#include<conio.h>

class Employee {

protected:

int eno;

char ename[30],dept[30],grade;

float bp;

public:

void getdata();

void putdata(); };

void Employee::getdata() {

cout<<"\n--------------------------------------------------------------------------------------";

cout<<"\n\t\t DEPT OF CA";

cout<<"\n\t\t Bharathiar University";

cout<<"\n\n\t\t\t EMPLOYEE PAYROLL SYSTEM";

cout<<"\n--------------------------------------------------------------------------------------";

cout<<"\nEnter Employee No: ";

cin>>eno;

cout<<"\nEnter Employee Name: ";

cin>>ename;

cout<<"\nEnter Employee Dept: ";

cin>>dept;

cout<<"\nEnter Employee BasicPay: ";

cin>>bp;

cout<<"\nEnter Employee Grade A or B or C: ";

cin>>grade;

cout<<"\n---------------------------------------------------------------------------------------";

}

void Employee::putdata() {

cout<<"\n---------------------------------------------------------------------------------------";

cout<<"\n\t\t DEPT OF CA";

cout<<"\n\t\t Bharathiar University";

cout<<"\n\n\t\t\t\t PAYSLIP";

cout<<"-----------------------------------------------------------------------------------------";

cout<<"\n EMPLOYEE NO:"<<eno;

cout<<"\t\t\t EMPLOYEE NAME:"<<ename;

cout<<"\n EMPLOYEE DEPT:"<<dept;

cout<<"\t\t\t\t EMPLOYEE GRADE:"<<grade;

cout<<"\n---------------------------------------------------------------------------------------";

}

class Pay: public Employee

{

float da,hra,pf,gp,np,ins;

public:

void call() {

if((grade=='A')||(grade=='a')) {

da=0.40\*bp;

hra=0.30\*bp;

pf=0.20\*bp;

gp=bp+da+hra;

ins=300;

np=gp-pf-ins;

getPayDetails();

}

else if((grade=='B')||(grade=='b')) {

da=0.35\*bp;

hra=0.25\*bp;

pf=0.15\*bp;

gp=bp+da+hra;

ins=250;

np=gp-pf-ins;

getPayDetails();

}

else if((grade=='C')||(grade=='c')) {

da=0.30\*bp;

hra=0.20\*bp;

pf=0.10\*bp;

gp=bp+da+hra;

ins=200;

np=gp-pf-ins;

getPayDetails();

}

else {

cout<<"\n\n\t\t INVALID GRADE";

cout<<"\n-----------------------------------------------------------";

} }

void getPayDetails() {

cout<<"\nALLOWANCE :\n";

cout<<"\n\tDAILY ALLOWANCE :"<<da<<".00";

cout<<"\n\tHR ALLOWANCE :"<<hra<<".00";

cout<<"\n--------------------------------------------------------------------";

cout<<"\nDEDUCTION: \n";

cout<<"\n\tPROVIDENT FUND :"<<pf<<".00";

cout<<"\n\tINSURANCE :"<<pf<<".00";

cout<<"\n\tNET PAY :"<<pf<<".00"; } };

int main(){

int ano=1;

Pay p;

do {

p.getdata();

p.putdata();

p.call();

cout<<"Press 1 to continue or 0 to Exit";

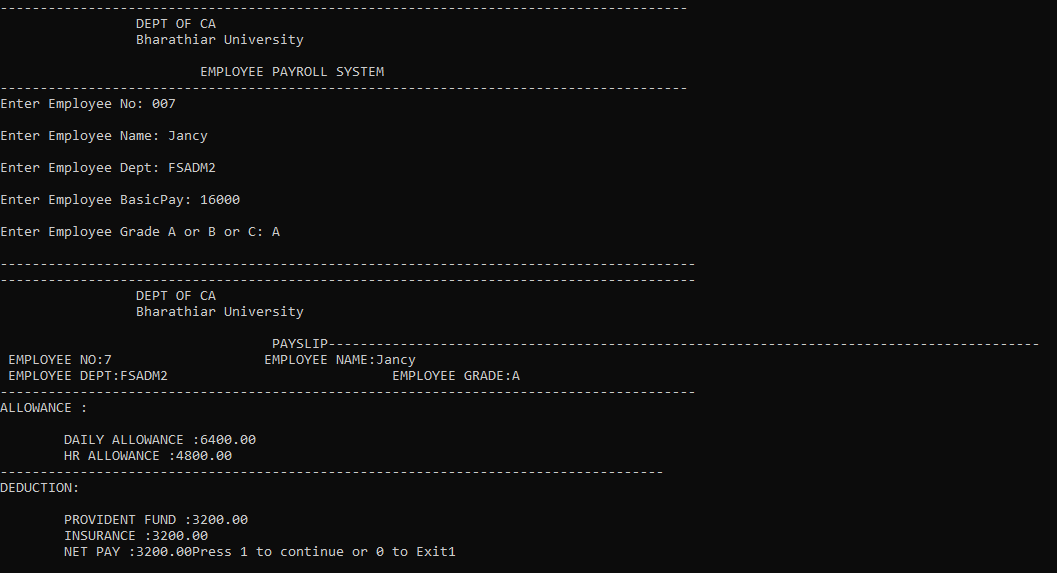
cin>>ano; }

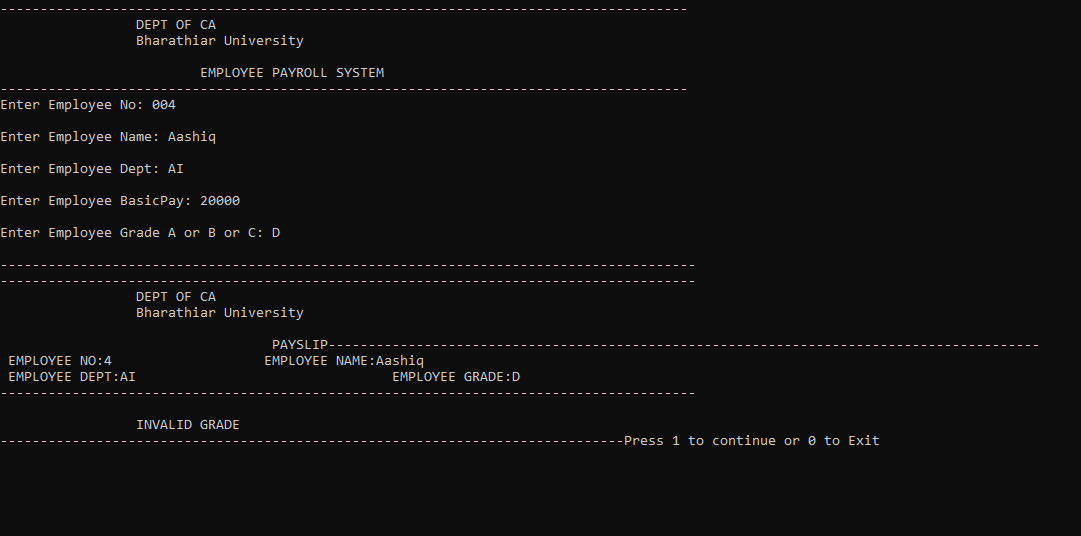
while(ano==1);

getch();

}

**Output:**





**Coding: (Electricity Bill Using Files)**

#include<iostream>

using namespace std;

#include<conio.h>

#include<fstream>

int main()

{

char ac[20],mon[10],sn[20];

int units;

long int cr,pr;

float amt;

ofstream fout;

fout.open("D:Files.TXT");

cout<<"\n-------------------------------------------------------------------------------------";

cout<<"\n\t\t ELECTRICITY BILL BY USING FILES";

cout<<"\n-------------------------------------------------------------------------------------";

cout<<"\n\nENTER AREA CODE :";

cin>>ac;

cout<<"\n\nENTER SERVICE NUMBER :";

cin>>sn;

cout<<"\n\nENTER CURRENT MONTH :";

cin>>mon;

cout<<"\n\nENTER CURRENT READING :";

cin>>cr;

cout<<"\n\nENTER PREVIOUS READING :";

cin>>pr;

units=cr-pr;

if(units<=100)

{

if(units>=1 && units<=50){

amt=units\*0.65;

}

else{

amt=units\*0.75;

}

}

else if(units>=101 && units<=600)

{

if(units>=101 && units<=400){

amt=units\*1.50;

}

else if(units>=201 && units<=400){

amt=units\*2.25;

}

else{

amt=units\*6;

}

}

if(amt<=40){

amt=40;

}

fout<<"\n-----------------------------------------------------------------------------";

fout<<"\n\t\tELECTRICITY BILL USING FILES ";

fout<<"\n-----------------------------------------------------------------------------";

fout<<"\n\n\t\t AREA CODE : "<<ac;

fout<<"\n\n\t\t SERVICE NUMBER : "<<sn;

fout<<"\n\n\t\t MONTH : "<<mon;

fout<<"\n\n\t\t CURRENT READING : "<<cr;

fout<<"\n\n\t\t PREVIOUS READING : "<<pr;

fout<<"\n-----------------------------------------------------------------------------";

fout.close();

fout.open("D:\\Receipt.txt");

fout<<"\n-----------------------------------------------------------------------------";

fout<<"\n\t\t TAMILNADU ELECTRICITY BOARD ";

fout<<"\n\t\t COIMBATORE REGIONAL OFFICE, COIMBATORE ";

fout<<"\n----------------------------------------------------------------------------";

fout<<"\n\n\t\t AREA CODE : "<<ac;

fout<<"\n\n\t\t SERVICE NUMBER : "<<sn;

fout<<"\n\n\t\t BILL MONTH : "<<mon;

fout<<"\n\n\t\t CURRENT READING : "<<cr;

fout<<"\n\n\t\t PREVIOUS READING : "<<pr;

fout<<"\n\n\t\t UNITS RECORDED : "<<units;

fout<<"\n\n\t\t AMOUNT PAYABLE : "<<amt;

fout<<"\n\n\t\t PAY DATE: 30th "<<mon<<"\tWITH FINE : 15th

NEXT MONTH";

fout<<"\n-------------------------------------------------------------------------------------";

fout.close();

ifstream fin;

cout<<"\n-------------------------------------------------------------------------------------";

cout<<"\n\t\t TAMILNADU ELECTRICITY BOARD ";

cout<<"\n\t\t COIMBATORE REGIONAL OFFICE, COIMBATORE ";

cout<<"\n-------------------------------------------------------------------------------------";

cout<<"\n\n\t\t AREA CODE : "<<ac;

cout<<"\n\n\t\t SERVICE NUMBER : "<<sn;

cout<<"\n\n\t\t BILL MONTH : "<<mon;

cout<<"\n\n\t\t CURRENT READING : "<<cr;

cout<<"\n\n\t\t PREVIOUS READING : "<<pr;

cout<<"\n\n\t\t UNITS RECORDED : "<<units;

cout<<"\n\n\t\t AMOUNT PAYABLE : "<<amt;

cout<<"\n\n\t\t PAY DATE: 30th "<<mon<<"\tWITH FINE : 15th NEXT MONTH";

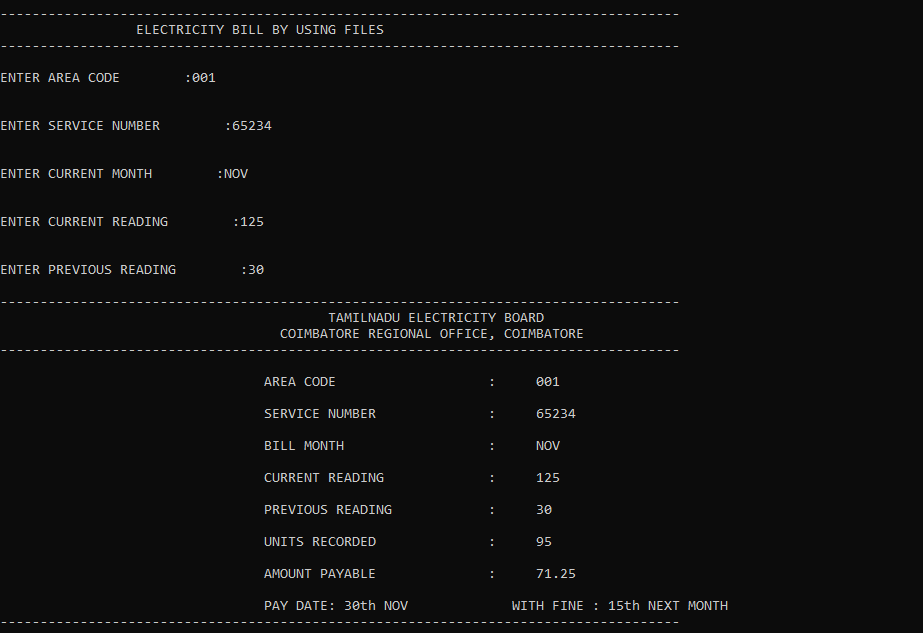
cout<<"\n-------------------------------------------------------------------------------------";

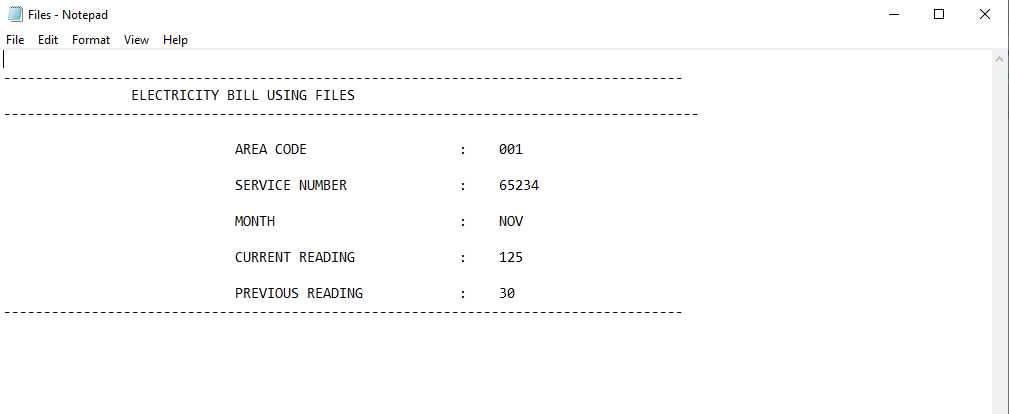
fin.close();

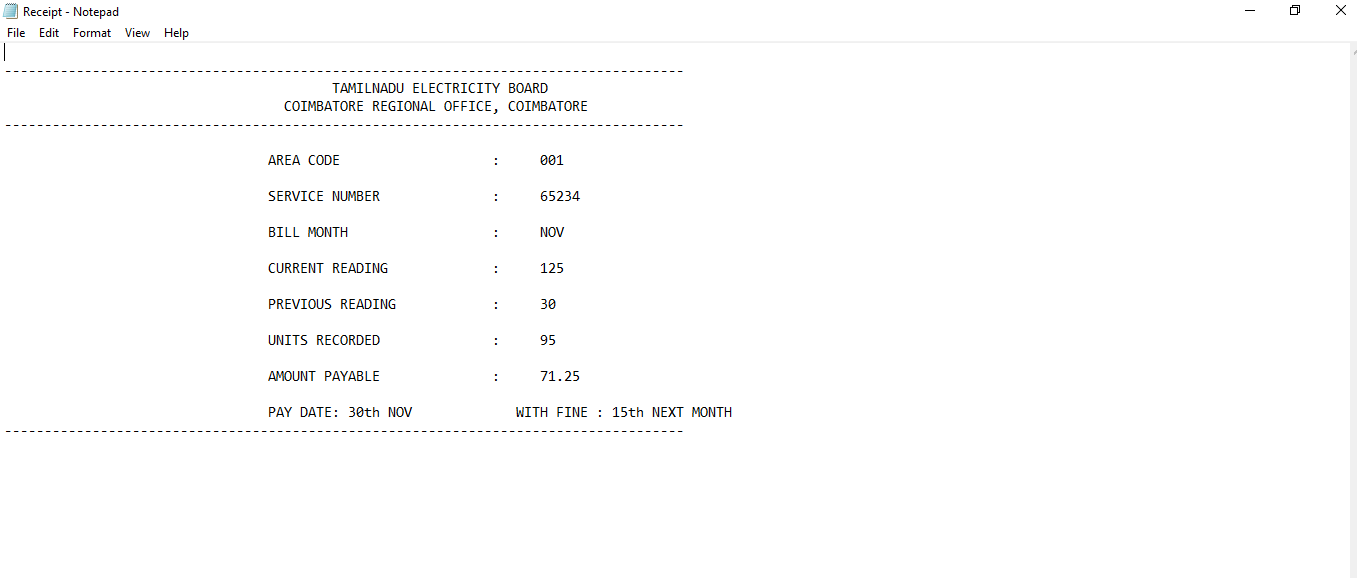
getch();

}

**Output:**







**Coding: (Finding area of rectangle and triangle using Polymorphism)**

#include<iostream>

using namespace std;

#include<conio.h>

class CPolygon

{

protected:

int width,height;

public:

void set\_values(int a, int b){

width=a;

height=b; }

};

class CRectangle : public CPolygon

{

public:

int area(){

return (width\*height); }

};

class CTriangle : public CPolygon

{

public:

int area(){

return (width\*height/2);

}

};

int main(){

int wid,heg;

cout<<"Enter the width :";

cin>>wid;

cout<<"Enter the height :";

cin>>heg;

CRectangle rect;

CTriangle trgl;

CPolygon \* poly1 = &rect;

CPolygon \* poly2 = &trgl;

poly1->set\_values(wid,heg);

poly2->set\_values(wid,heg);

cout<<"\n\n RESULT "<<endl;

cout<<"~~~~~~~~~~~~~"<<endl;

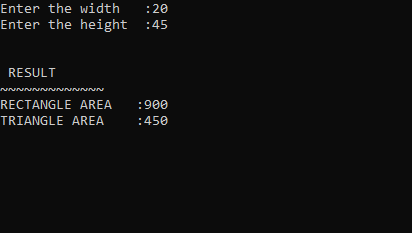
cout<<"RECTANGLE AREA :"<<rect.area()<<endl;

cout<<"TRIANGLE AREA :"<<trgl.area()<<endl;

getch();

return 0; }

**Output:**



**Coding: //Polymorphism**

class Employee

{

Employee (String n,double s)

{

name=n;

salary=s;

}

public void setSalary(double s)

{

salary=s;

}

public String getName()

{

return name;

}

public double getSalary()

{

return salary;

}

private String name;

private double salary;

}

class Manager extends Employee

{

Manager(String s, double d, double b)

{

super(s,d);

bonus=b;

}

public void setBonus(double b)

{

bonus=b;

}

public double getBonus()

{

return bonus;

}

public double getSalary() //method overloading.

{

double basesalary =super.getSalary(); //calls getSalary method.

return basesalary+bonus;

}

private double bonus;

}

class Polymorphism

{

public static void main(String[]args)

{

Employee e =new Employee("Programmer",20000);

Manager m= new Manager("SBI",20000,5000);

e= new Manager("SBI",20000,5000);

System.out.println("Bank Name :" +e.getName());

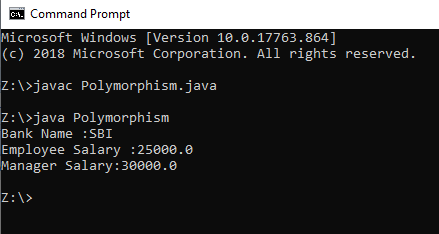
System.out.println("Employee Salary :" +e.getSalary());

System.out.println("Manager Salary:" +(m.getBonus()+e.getSalary()));

}

}

**Output:**



**Coding: (Adding Two Numbers)**

public class AddTwoNumbers

{

public static void main (String[] args)

{

int num1=5, num2=15, sum;

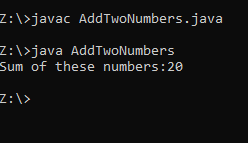
sum = num1+num2;

System.out.println("Sum of these numbers:" +sum);

}

}

**Output:**



**Coding: (adding Using Two Numbers using scanner)**

import java.util.Scanner;

public class AddTwoNumbers2

{

public static void main (String[] args)

{

int num1, num2, sum;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Number:");

num1 = sc.nextInt();

System.out.println("Enter the Second Number:");

num2 = sc.nextInt();

sc.close();

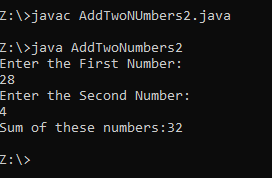
sum = num1+num2;

System.out.println("Sum of these numbers:" +sum);

}

}

Output:



**Coding: (Mouse Operations)**

/\* MOUSE EVENTS \*/

import java.io.\*;

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

/\*

<applet code="Mouse.class" width=300 height=300>

</applet>

\*/

public class Mouse extends Applet implements MouseListener, MouseMotionListener

{

String txt = "Welcome to MCA";

int x=10, y=30;

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

public void mouseClicked(MouseEvent me)

{

txt="Mouse Clicked";

setForeground(Color.pink);

repaint();

}

public void mouseEntered(MouseEvent me)

{

txt="Mouse Entered";

//setForeground(Color.green);

repaint();

}

public void mouseExited(MouseEvent me)

{

txt="Mouse Exited";

setForeground(Color.blue);

repaint();

}

public void mousePressed(MouseEvent me)

{

txt="Mouse Pressed";

setForeground(Color.yellow);

repaint();

}

public void mouseMoved(MouseEvent me)

{

txt="Mouse Moved";

setForeground(Color.red);

repaint();

}

public void mouseDragged(MouseEvent me)

{

txt="Mouse Dragged";

setForeground(Color.green);

repaint();

}

public void mouseReleased(MouseEvent me)

{

txt="Mouse Released";

setForeground(Color.red);

repaint();

}

public void paint(Graphics g)

{

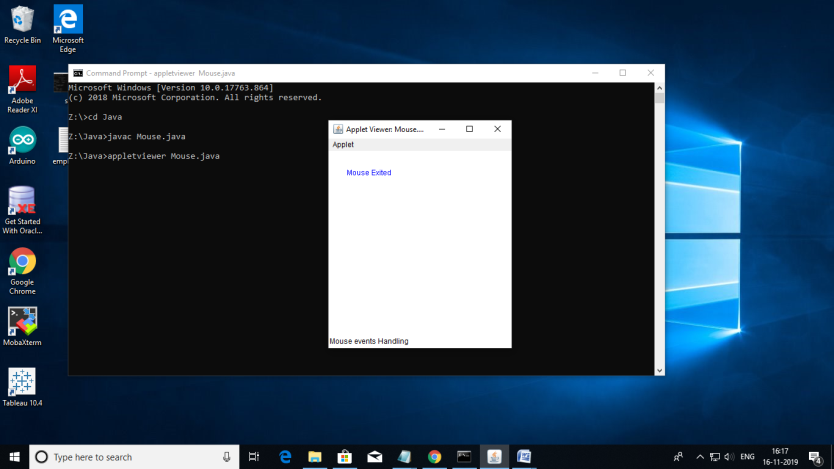
g.drawString(txt,30,40);

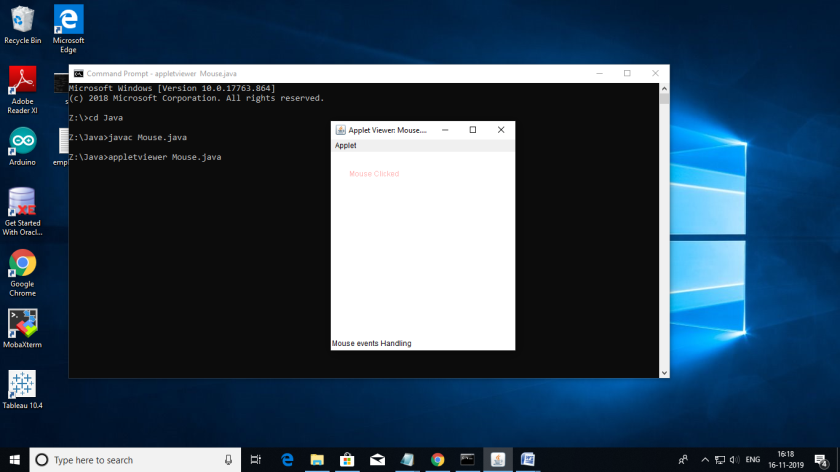
showStatus("Mouse events Handling");

}

}

**Output:**





**Coding: (Calculator using applet)**

/\* CALCULATOR \*/

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

/\*

<applet code="Calc.class" width=300 height=300>

</applet>

\*/

public class Calc extends Applet implements ActionListener

{

String msg="";

int v1,v2,result;

TextField t1;

Button b[]=new Button[10];

Button add, sub, mul, div, clear, mod, EQ;

char OP;

public void init() {

Color k = new Color(120, 89, 90);

setBackground(k);

t1=new TextField(10);

GridLayout g1 = new GridLayout(10, 5);

setLayout(g1);

GridLayout g2 = new GridLayout(4, 5);

setLayout(g2);

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

{

b[i] = new Button(""+i);

}

add = new Button("add");

sub = new Button("sub");

mul = new Button("mul");

div = new Button("div");

mod = new Button("mod");

clear = new Button("clear");

EQ = new Button("EQ");

t1.addActionListener(this);

add(t1);

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

{

add(b[i]);

}

add(add);

add(sub);

add(mul);

add(div);

add(mod);

add(clear);

add(EQ);

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

{

b[i].addActionListener(this);

}

add.addActionListener(this);

sub.addActionListener(this);

mul.addActionListener(this);

div.addActionListener(this);

mod.addActionListener(this);

clear.addActionListener(this);

EQ.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String str = ae.getActionCommand();

char ch = str.charAt(0);

if(Character.isDigit(ch))

t1.setText(t1.getText()+str);

else

if(str.equals("add"))

{

v1=Integer.parseInt(t1.getText());

OP='+';

t1.setText("");

}

else if(str.equals("sub"))

{

v1=Integer.parseInt(t1.getText());

OP='-';

t1.setText("");

}

else if(str.equals("mul"))

{

v1=Integer.parseInt(t1.getText());

OP='\*';

t1.setText("");

}

else if(str.equals("div"))

{

v1=Integer.parseInt(t1.getText());

OP='/';

t1.setText("");

}

else if(str.equals("mod"))

{

v1=Integer.parseInt(t1.getText());

OP='%';

t1.setText("");

}

else if(str.equals("EQ"))

{

v2=Integer.parseInt(t1.getText());

if(OP=='+')

result=v1+v2;

else if(OP=='-')

result=v1-v2;

else if(OP=='\*')

result=v1\*v2;

else if(OP=='/')

result=v1/v2;

else if(OP=='%')

result=v1%v2;

t1.setText(""+result);

}

if(str.equals("clear"))

{

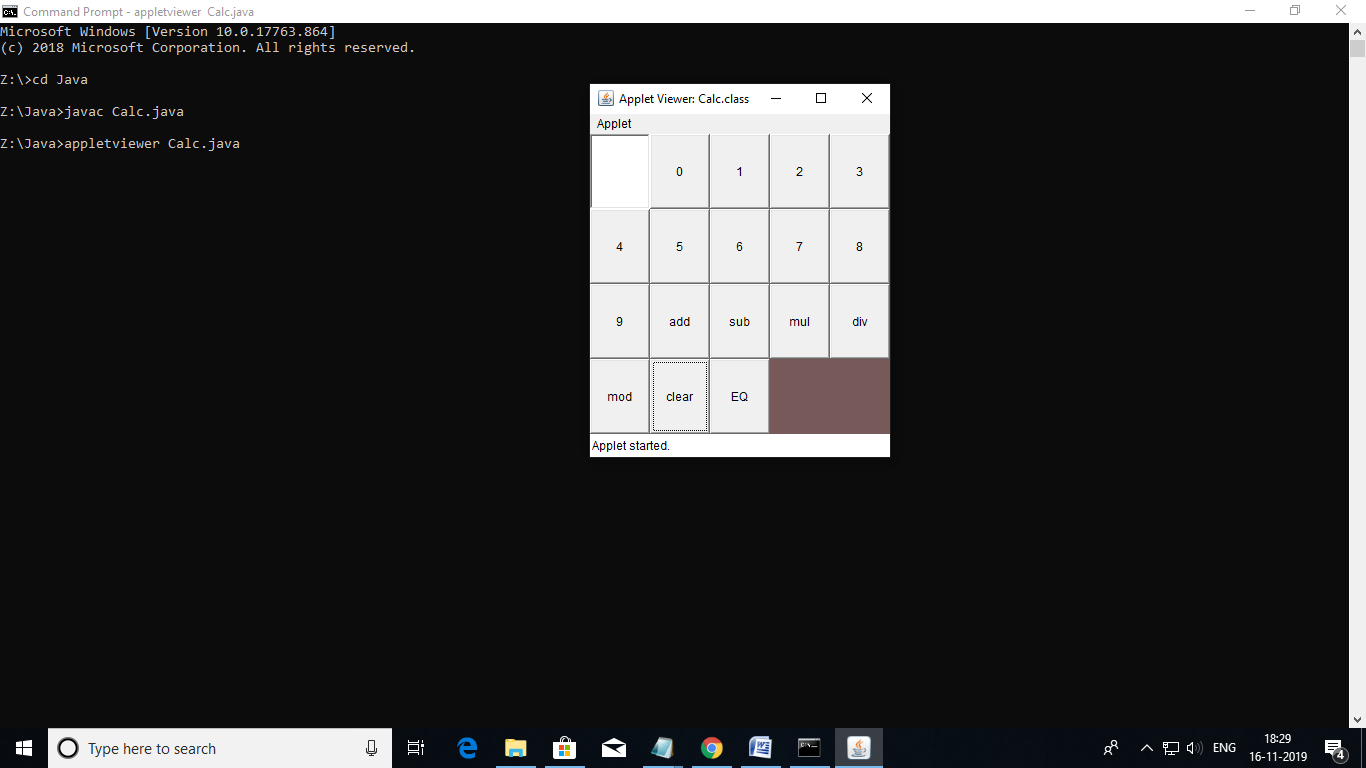
t1.setText("");

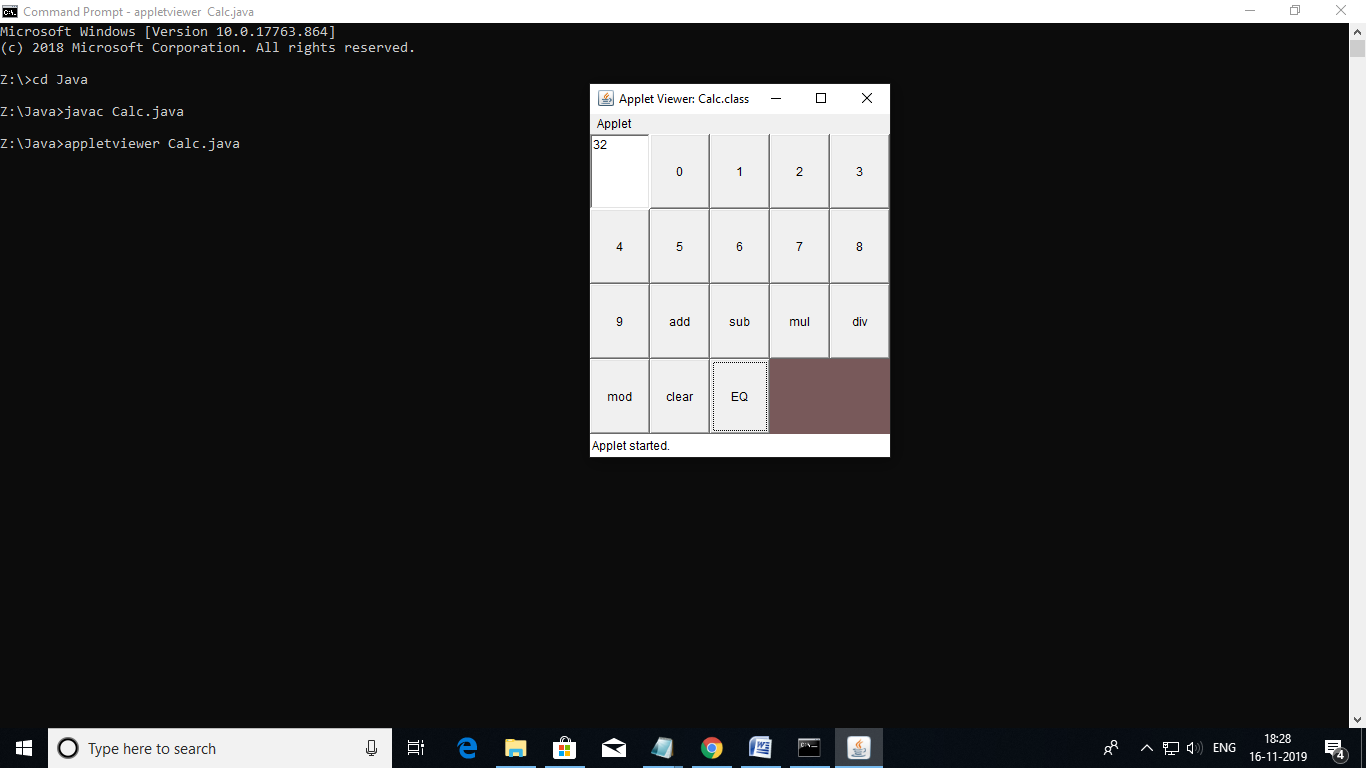
}

}

}

**Output:**





**Coding: //Digital Clock**

import java.applet.Applet;

import java.awt.\*;

import java.util.\*;

import java.text.\*;

/\*

<applet code="DigitalClock.class" width=300 height=300>

</applet>

\*/

public class DigitalClock extends Applet implements Runnable {

Thread t;

String Time="";

public void init()

{ setSize(400,200);

setBackground(Color.pink);

t=new Thread(this);

t.start();

}

public void run()

{

try

{

while(true)

{

Calendar c=Calendar.getInstance();

SimpleDateFormat f=new SimpleDateFormat("hh:mm:ss "+" dd/MM/yyyy");

Date d=c.getTime();

Time=f.format(d);

repaint();

t.sleep(1000);

}

}catch(Exception e)

{

System.out.println("Error Occured");

}

}

public void paint(Graphics g)

{

g.setColor(Color.RED);

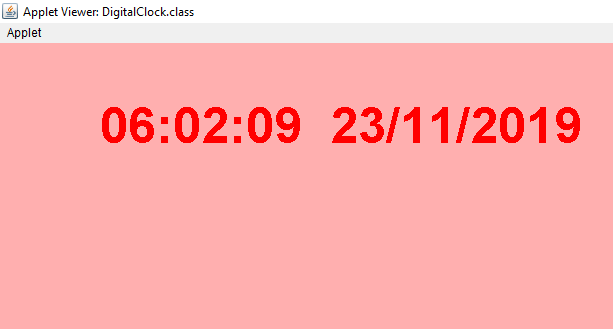
g.setFont(new Font("Dialog",Font.BOLD,50));

g.drawString(Time, 100,100);

}

}

**Output:**



**Coding: //Default Mutable Tree Node**

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import javax.swing.\*;

import javax.swing.tree.\*;

public class pg9 extends JApplet

{

DefaultMutableTreeNode course;

DefaultMutableTreeNode pg;

DefaultMutableTreeNode ug;

DefaultMutableTreeNode dip;

DefaultMutableTreeNode pg1,pg2,pg3;

DefaultMutableTreeNode pg31,pg32,pg33;

DefaultMutableTreeNode ug1,ug2,ug3;

DefaultMutableTreeNode dip1,dip2,dip3;

JTextField jtf;

JTree jtree;

public void init()

{

Container con=getContentPane();

con.setLayout(new BorderLayout());

jtf=new JTextField(40);

course=new DefaultMutableTreeNode("course");

pg=new DefaultMutableTreeNode("Post Graduate courses");

ug=new DefaultMutableTreeNode("Under Graduate courses");

dip=new DefaultMutableTreeNode("Diploma courses");

pg1=new DefaultMutableTreeNode("M.C.A");

pg2=new DefaultMutableTreeNode("M.B.A");

pg3=new DefaultMutableTreeNode("M.sc");

pg31=new DefaultMutableTreeNode("M.Sc(IT)");

pg32=new DefaultMutableTreeNode("M.Sc(CS)");

pg33=new DefaultMutableTreeNode("M.sc(Maths)");

ug1=new DefaultMutableTreeNode("B.C.A");

ug2=new DefaultMutableTreeNode("B.B.A");

ug3=new DefaultMutableTreeNode("B.sc");

dip1=new DefaultMutableTreeNode("P.G.D.C.A");

dip2=new DefaultMutableTreeNode("P.G.D.F.T");

dip3=new DefaultMutableTreeNode("P.G.D.I.D");

pg.add(pg1);

pg.add(pg2);

pg.add(pg3);

pg3.add(pg31);

pg3.add(pg32);

pg3.add(pg33);

ug.add(ug1);

ug.add(ug2);

ug.add(ug3);

dip.add(dip1);

dip.add(dip2);

dip.add(dip3);

course.add(pg);

course.add(ug);

course.add(dip);

jtree=new JTree(course);

int vscroll=ScrollPaneConstants.VERTICAL\_SCROLLBAR\_ALWAYS;

int hscroll=ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_ALWAYS;

JScrollPane jsp=new JScrollPane(jtree,vscroll,hscroll);

con.add(jsp,BorderLayout.CENTER);

con.add(jtf,BorderLayout.SOUTH);

jtree.addMouseListener(new MouseAdapter()

{

public void mouseClicked(MouseEvent me)

{

TreePath tp=jtree.getSelectionPath();

if(tp!=null)

jtf.setText(" "+tp);

else

jtf.setText(" ");

}

});

}

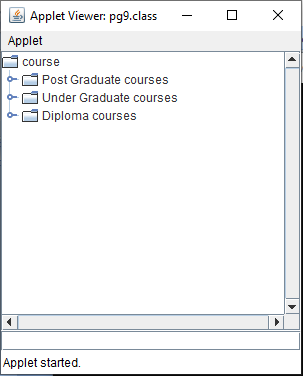
}

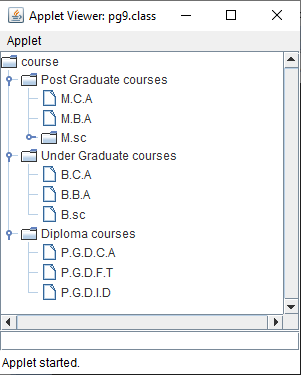
/\*<applet code="pg9.class" width=300 height=300>

</applet>

\*/

**Output:**

****

****

**Coding: //Student Records**

import java.io.\*;

class StudentRecords

{

static BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

public void addRecords() throws IOException

{

// Create or Modify a file for Database

PrintWriter pw = new PrintWriter(new BufferedWriter(new

FileWriter("studentRecords.txt",true)));

String name, Class, fname, mname, address, dob;

int age;

long telephoneNo;

String s;

boolean addMore = false;

// Read Data

do

{

System.out.print("\nEnter name: ");

name = br.readLine();

System.out.print("Father's Name: ");

fname = br.readLine();

System.out.print("Mother's Name: ");

mname = br.readLine();

System.out.print("Address: ");

address = br.readLine();

System.out.print("Class: ");

Class = br.readLine();

System.out.print("Date of Birth (dd/mm/yy) : ");

dob = br.readLine();

System.out.print("Age: ");

age = Integer.parseInt(br.readLine());

System.out.print("Telephone No.: ");

telephoneNo = Long.parseLong(br.readLine());

// Print to File

pw.println(name);

pw.println(fname);

pw.println(mname);

pw.println(address);

pw.println(Class);

pw.println(dob);

pw.println(age);

pw.println(telephoneNo);

System.out.print("\nRecords added successfully !\n\nDo you want to add more records ? (y/n) : ");

s = br.readLine();

if(s.equalsIgnoreCase("y"))

{

addMore = true;

System.out.println();

}

else

addMore = false;

}

while(addMore);

pw.close();

showMenu();

}

public void readRecords() throws IOException

{

try

{

// Open the file

BufferedReader file = new BufferedReader(new

FileReader("studentRecords.txt"));

String name;

int i=1;

// Read records from the file

while((name = file.readLine()) != null)

{

System.out.println("S.No. : " +(i++));

System.out.println("-------------");

System.out.println("\nName: " +name);

System.out.println("Father's Name : "+file.readLine());

System.out.println("Mother's Name : "+file.readLine());

System.out.println("Address: "+file.readLine());

System.out.println("Class: "+file.readLine());

System.out.println("Date of Birth : "+file.readLine());

System.out.println("Age: "+Integer.parseInt(file.readLine()));

System.out.println("Tel. No.: "+Long.parseLong(file.readLine()));

System.out.println();

}

file.close();

showMenu();

}

catch(FileNotFoundException e)

{

System.out.println("\nERROR : File not Found !!!");

}

}

public void clear() throws IOException

{

// Create a blank file

PrintWriter pw = new PrintWriter(new BufferedWriter(new

FileWriter("studentRecords.txt")));

pw.close();

System.out.println("\nAll Records cleared successfully !");

for(int i=0;i<999999999;i++); // Wait for some time

showMenu();

}

public void showMenu() throws IOException

{

System.out.print("1 : Add Records\n2 : Display Records\n3 : Clear All Records\n4 : Exit\n\nYour Choice : ");

int choice = Integer.parseInt(br.readLine());

switch(choice)

{

case 1:

addRecords();

break;

case 2:

readRecords();

break;

case 3:

clear();

break;

case 4:

System.exit(1);

break;

default:

System.out.println("\nInvalid Choice !");

break;

}

}

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

{

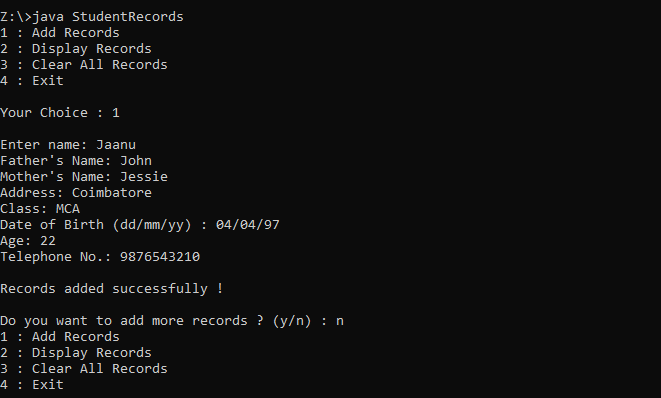
StudentRecords call = new StudentRecords();

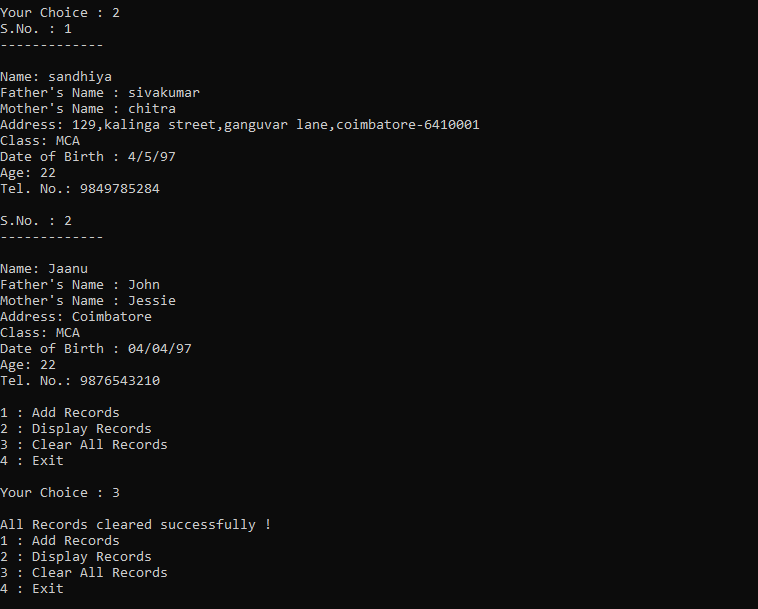
call.showMenu();

}

}

**Output:**

****

****