|  |  |
| --- | --- |
| **2 . Write a program to pass Starting and Ending limit and print all prime numbers and Fibonacci numbers between this range**. |  |

**CODE:**

import java.util.\*;

public class prime

{

public static void main (String args[])

{

int start,end,i,j,flag=0,k,a,b,c;

Scanner in = new Scanner(System.in);

System.out.println("Enter start number");

start = in.nextInt();

System.out.println("Enter end limit");

end = in.nextInt();

System.out.println("Prime number is :");

for(i=start;i<end;i++)

{

for(j=2;j<i;j++)

{

if(i%j == 0)

{

flag=0;

break;

}

else

{

flag=1;

}

}

if(flag == 1)

{

System.out.println(i);

}

}

System.out.println("Fibbonacci Series is");

a=start;

b=start +1;

System.out.println(a);

System.out.println(b);

for(k=start;k<=end;k++)

{

c=a+b;

if(c<=end)

{

System.out.println(c);

}

a=b;

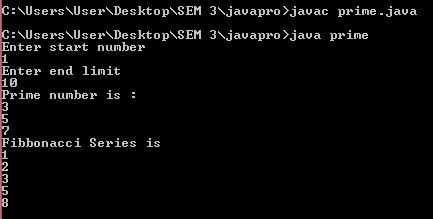
b=c;

}

}

}

**Output:**



|  |  |
| --- | --- |
| **3. Write a java program to check palindrome number. Input: 329 Output: not palindrome number**  **Input: 12321 Output: palindrome number**  **CODE:**  **import java.util.\*;**  **public class palindrome**  **{**  **public static void main(String args[])**  **{**  **int no,rev=0,rem=0,orig;**  **System.out.println("Enter number");**  **Scanner in = new Scanner(System.in);**  **no=in.nextInt();**  **orig=no;**  **while(no > 0)**  **{**  **rem=no%1**  **rev=rev\*10+rem;**  **no=no/10;**  **}**  **if(rev==orig)**  **{**  **System.out.println(orig + "Palindrome number");**  **}**  **else**  **{**  **System.out.println(orig + "Not Palindrome number");**  **}**  **}**  **}**  **Output:-** |  |
| **4. Write a java program to print factorial of a number.Input: 5 Output: 120**  **Input: 6 Output: 720**  **CODE:**  **import java.util.\*;**  **public class fact**  **{**  **public static void main(String args[])**  **{**  **int fact=1,no;**  **System.out.println("Enter number");**  **Scanner in = new Scanner(System.in);**  **no=in.nextInt();**  **while(no > 0)**  **{**  **fact=fact\*no;**  **no--;**  **}**  **System.out.println("Factorial is" + fact);**  **}**  **}**  **Output:** |  |

|  |  |
| --- | --- |
| **5. Write a java program to check Armstrong number.**  **Input: 153 Output: Armstrong number**  **Input: 22 Output: not Armstrong number**  **CODE:**  **import java.util.\*;**  **class arm**  **{**  **public static void main(String args[])**  **{**  **int rem=0,sum=0,no,orig;**  **Scanner in = new Scanner(System.in);**  **System.out.println("Enter no");**  **no=in.nextInt();**  **orig=no;**  **while(no>0)**  **{**  **rem=no%10;**  **sum=rem\*rem\*rem+sum;**  **no=no/10;**  **}**  **if(sum == orig)**  **{**  **System.out.println(orig + " is armstrong number");**  **}**  **else**  **{**  **System.out.println(orig + " is not armstrong number");**  **}**  **}**  **}**  **Output:** |  |

|  |  |
| --- | --- |
|  |  |

**6. Write a program in Java to find maximum of three numbers using conditiona operator**

**CODE**:

import java.util.\*;

public class max

{

public static void main(String [ ] args )

{

int a,b,c,max,temp;

System.out.println("Enter value of a");

Scanner in = new Scanner(System.in);

a=in.nextInt();

System.out.println("Enter value of b");

b=in.nextInt();

System.out.println("Enter value of c");

c=in.nextInt();

temp=(a>b)?a:b;

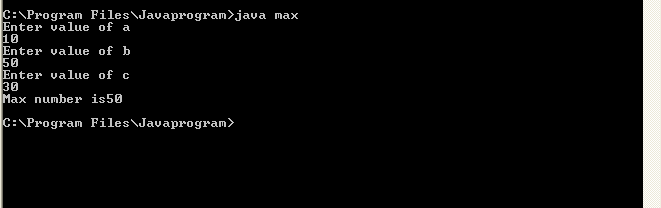
max= (c>temp)?c:temp;

System.out.println("Max number is"+max);

}

}

**OUTPUT**:



|  |
| --- |
| **7. Write a java program which should display maximum and minimum number of given 3 numbers.** |

**CODE**:

import java.util.\*;

public class maxmin

{

public static void main(String [ ] args )

{

int a,b,c,max,temp;

System.out.println("Enter value of a");

Scanner in = new Scanner(System.in);

a=in.nextInt();

System.out.println("Enter value of b");

b=in.nextInt();

System.out.println("Enter value of c");

c=in.nextInt();

if(a>b&&a>c&& b<c)

{

System.out.println("Max"+a);

System.out.println("Min"+b);

}

else if (b>a&&b>c&&c<a)

{

System.out.println("Max"+b);

System.out.println("Min"+c);

}

else if (c>a&&c>b)

{

System.out.println("Max"+c);

}

else

{

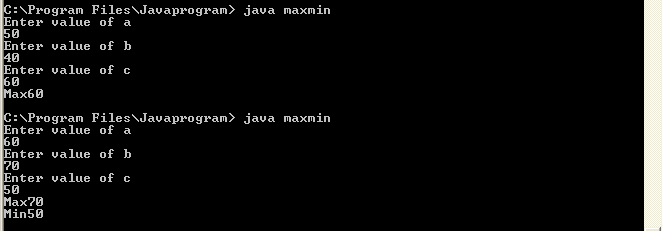
System.out.println("program");

}

}

}

**OUTPUT**:



**8. Write a program in Java to multiply two matrix**

**CODE:**

import java.util.Scanner;

class MatrixMultiplication

{

public static void main(String args[])

{

int m, n, p, q, sum = 0, c, d, k;

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows and columns of first matrix");

m = in.nextInt();

n = in.nextInt();

int first[][] = new int[m][n];

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

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

first[c][d] = in.nextInt();

System.out.println("Enter the number of rows and columns of

second matrix");

p = in.nextInt();

q = in.nextInt();

if (n != p)

System.out.println("The matrices can't be multiplied with each

other.");

else

{

int second[][] = new int[p][q];

int multiply[][] = new int[m][q];

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

for (c = 0; c < p; c++)

for (d = 0; d < q; d++)

second[c][d] = in.nextInt();

for (c = 0; c < m; c++)

{

for (d = 0; d < q; d++)

{

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

{

sum = sum + first[c][k]\*second[k][d];

}

multiply[c][d] = sum;

sum = 0;

}

}

System.out.println("Product of the matrices:");

for (c = 0; c < m; c++)

{

for (d = 0; d < q; d++)

System.out.print(multiply[c][d]+"\t");

System.out.print("\n")

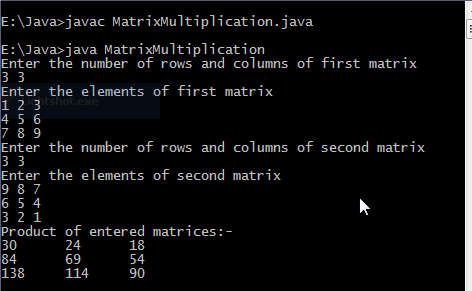
}

}

}

}

**Output:**



|  |
| --- |
| **9. Write a java program to create a class “Matrix” that would contain integer values having varied numbers of columns for each row. Print row-wise sum of the integer values for each row**. |

**CODE**:

public class Matrix3

{

public static void main(String args[])

{

int matrix[][]=new int[2][];

int sum[]=new int[5];

matrix[0]=new int[3];

matrix[1]=new int[2];

matrix[0][0]=14;

matrix[0][1]=22;

matrix[0][2]=32;

matrix[1][0]=40;

matrix[1][1]=50;

for(int i=0;i<2;i++)

{

for(int j=0;j<matrix[i].length;j++)

{

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

sum[i] = sum[i]+ matrix[i][j];

}

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

}

for(int x=0;x<2;x++)

{

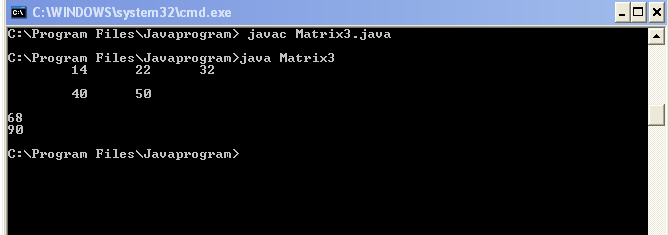
System.out.println(sum[x]);

}

}

}

**Output:**



|  |
| --- |
| **10. Write a Java application which takes several command line arguments, which are supposed to be names of students and prints output as given below:**  **(Suppose we enter 3 names then output should be as follows)..**  **Number of arguments = 3**  **1.: First Student Name is = Tom**  **2.: Second Student Name is = Dick** |

**CODE**:

public class command

{

public static void main(String [ ] args)

{

int i;

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

{

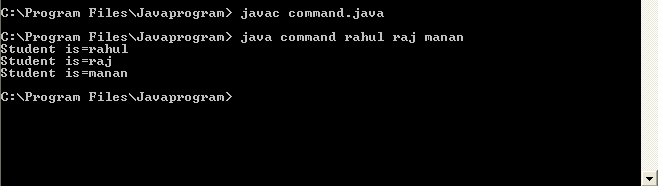
System.out.println("Student is="+args[i]);

}

}

}

**OUTPUT:**

****

**11. Write a Java application to count and display frequency of letters and digits from the String given by user as command-line argument.**

**CODE**:

public class eleven

{

public static void main(String[] args)

{

String str="i've scored 60% in sem3";

int fc=0;

int fn=0;

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

{

if(Character.isDigit(str.charAt(i)))

{

fn=fn+1;

}

else if(Character.isLetter(str.charAt(i)))

{

fc=fc+1;

}

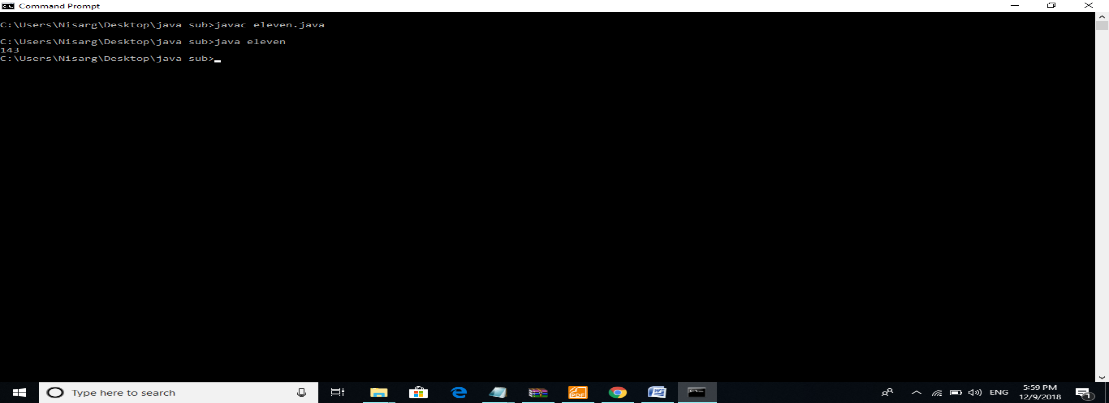
}

System.out.print(fc);

System.out.print(fn);

}

}

**Output:**

**12. Create a class “Student” that would contain enrolment No, name, and gender as data members. Create appropriate getter and setter methods for the “Student” class and constructor to initialize the data members. Also demonstrate constructor chaining.**

**CODE**:

public class Student

{

private int enrollmentNo;

private char gender;

private String name;

// Getter method always returns the value of a variable declared within the class

public int getnum()

{

return this.enrollmentNo;

}

public char getgen()

{

return this.gender;

}

public String getname()

{

return this.name;

}

public void setnum(int num)

{

enrollmentNo=num;

}

public void setgen(char g)

{

gender=g;

}

public void setname(String name)

{

this.name=name;

}

}

public class testStudent

{

public static void main(String args[])

{

Student Balaji=new Student();

Balaji.setnum(121);

Balaji.setgen('M');

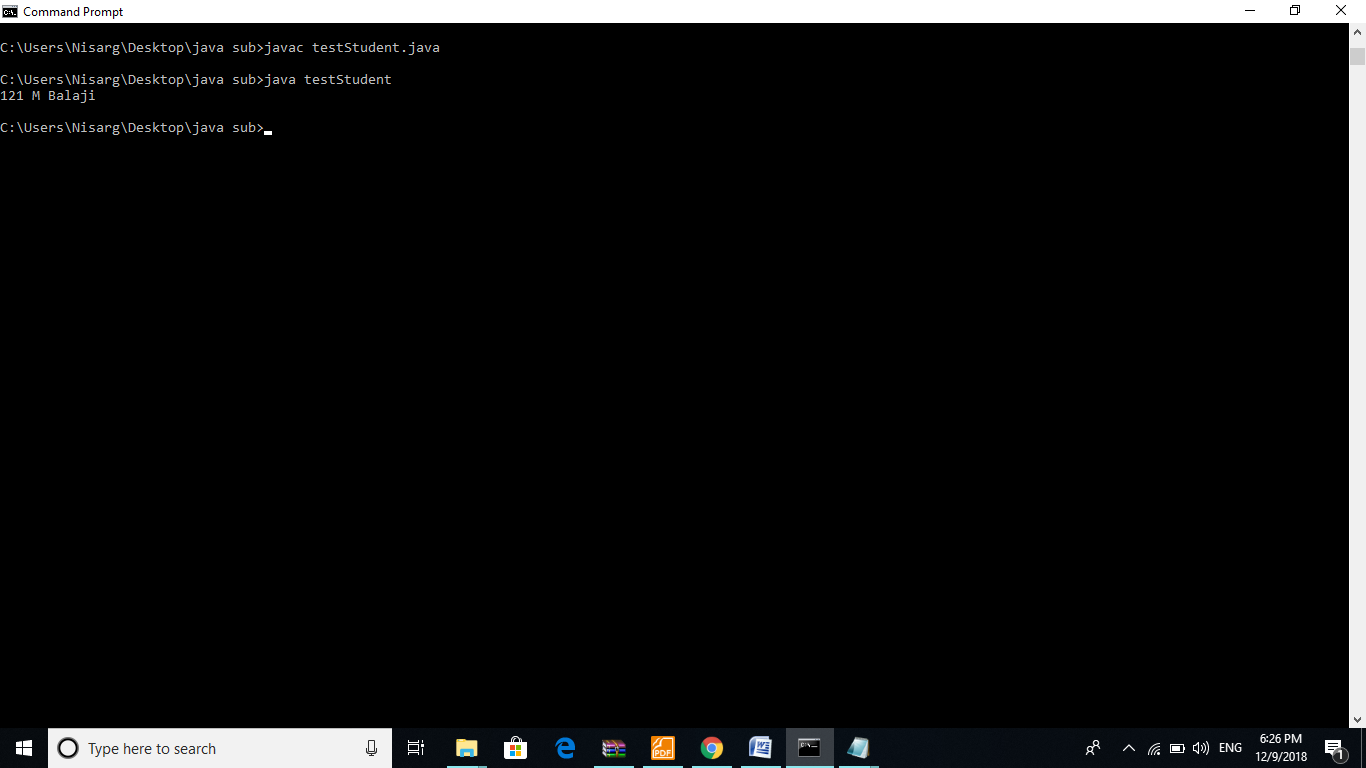
Balaji.setname("Balaji");

System.out.println(Balaji.getnum() + " " +Balaji.getgen() + " "+ Balaji.getname());

}

}

**Output**:



**13. Write a program in Java to demonstrate use of this keyword. Check whether this can access the private members of the class or not. [Refer class student in Q12 to perform the task]**

**CODE**:

class Test

{

private int n;

void setdata(int n)

{

this.n = n; //accessing private member (n) using this keyword.

}

void getdata()

{

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

}

}

class Example

{

public static void main(String[] args)

{

Test obj1 = new Test();

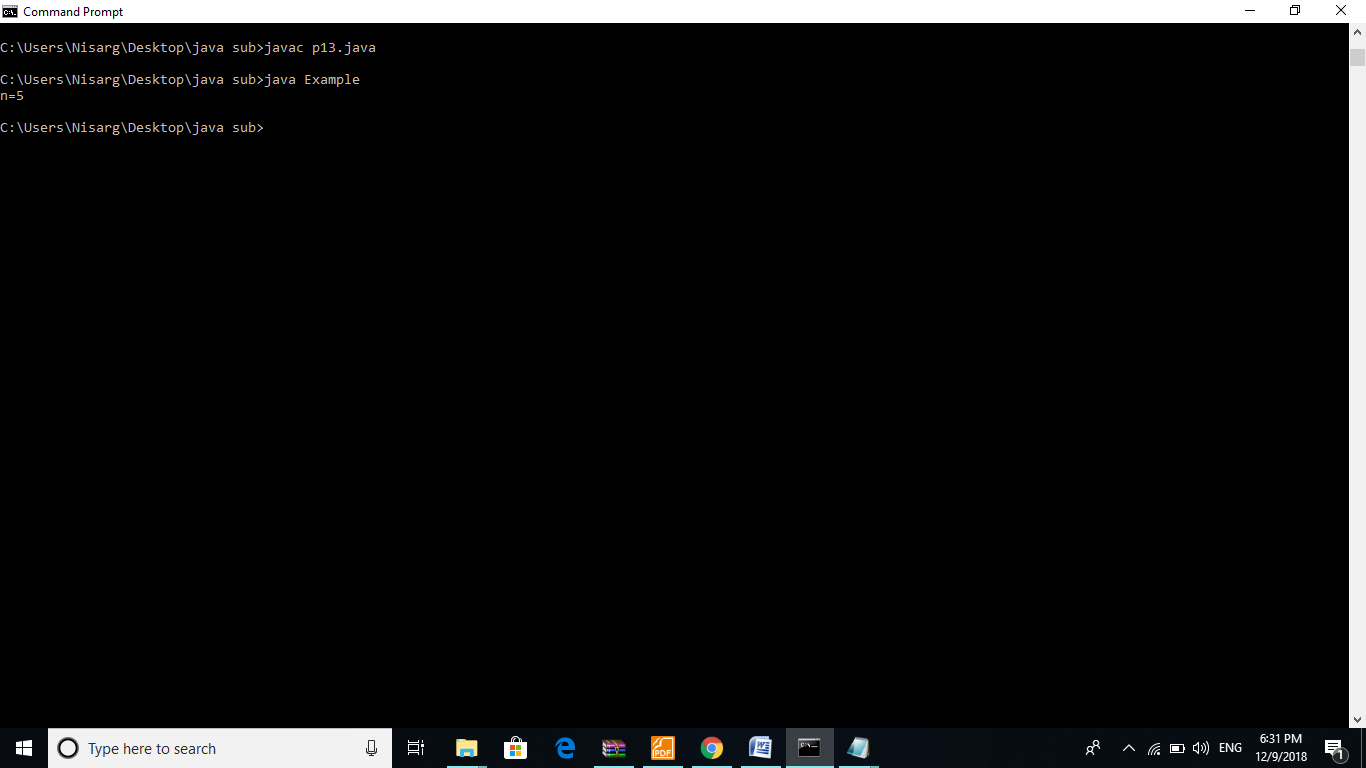
obj1.setdata(5);

obj1.getdata();

}

}

**Output:**



**14: Create a class “Rectangle” that would contain length and width as data members.**

**Define constructors [constructor overloading (default, parameterized and copy)] to initialize**

**the data members. Define the member functions to find area and to display the number objects**

**created.**

**[Note:define initializer block, static initializer block and the static data member and member function. Also demonstrate the sequence of execution of initializer block and static initialize block]**

**CODE**:

class Box

{

int length;

int width;

int height;

Box()

{

System.out.println("Inside Default Constructor");

length = 10;

width = 10;

height = 10;

}

Box(int l, int w,int h)

{

System.out.println("Inside Parameterized Constructor");

length = l;

width = w;

height = h;

}

Box(Box b)

{

System.out.println("Inside Copy Constructor");

length = b.length;

width = b.width;

height = b.height;

}

void volume()

{

int vol = length \* width \* height;

System.out.println("Volume of the Box: " + vol );

}

}

class Example

{

public static void main(String[] args)

{

Box b1= new Box();

b1.volume();

Box b2= new Box(5,5,10);

b2.volume();

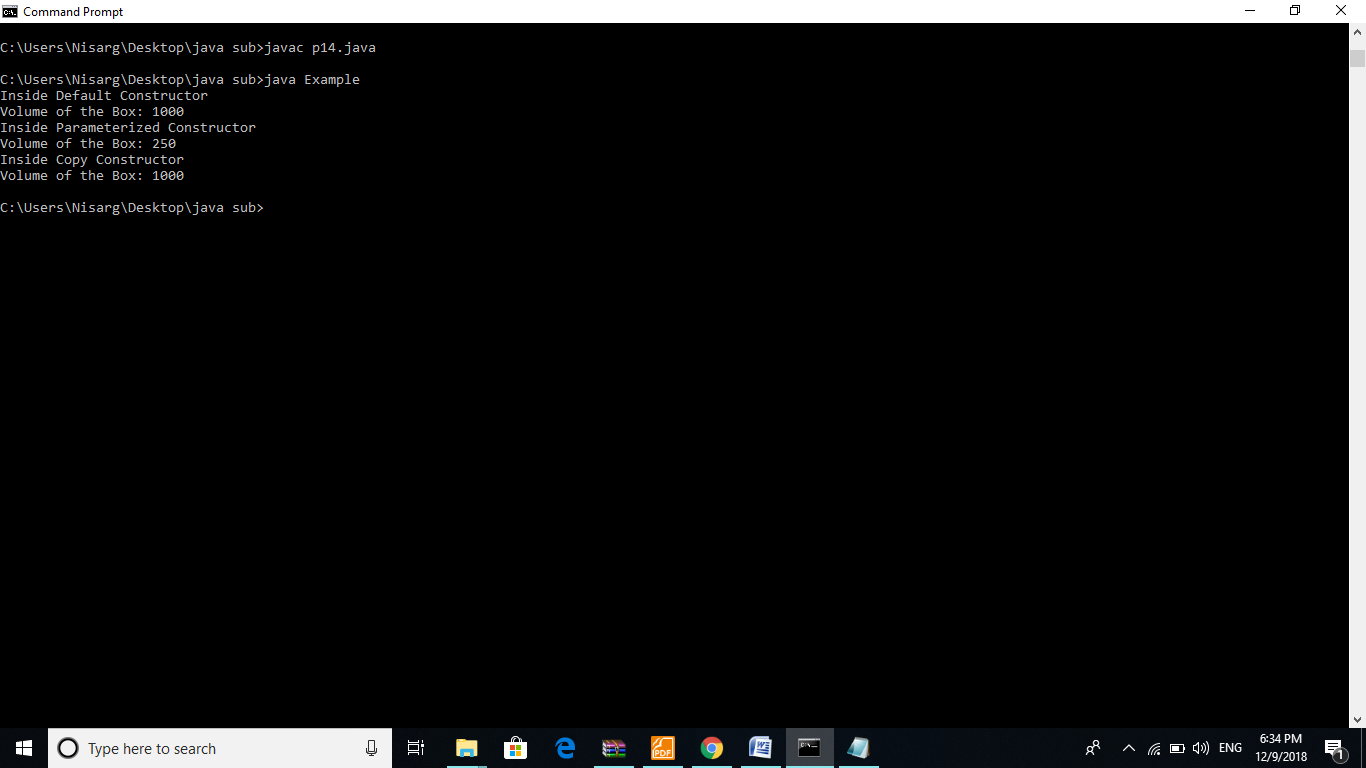
Box b3= new Box(b1);

b3.volume();

}

}

**Output:**



**15. Write a java program static block which will be executed before main ( ) method in a class.**

**CODE**:

public class java15

{

static String str;

static

{

str="Hello world";

System.out.println(str);

}

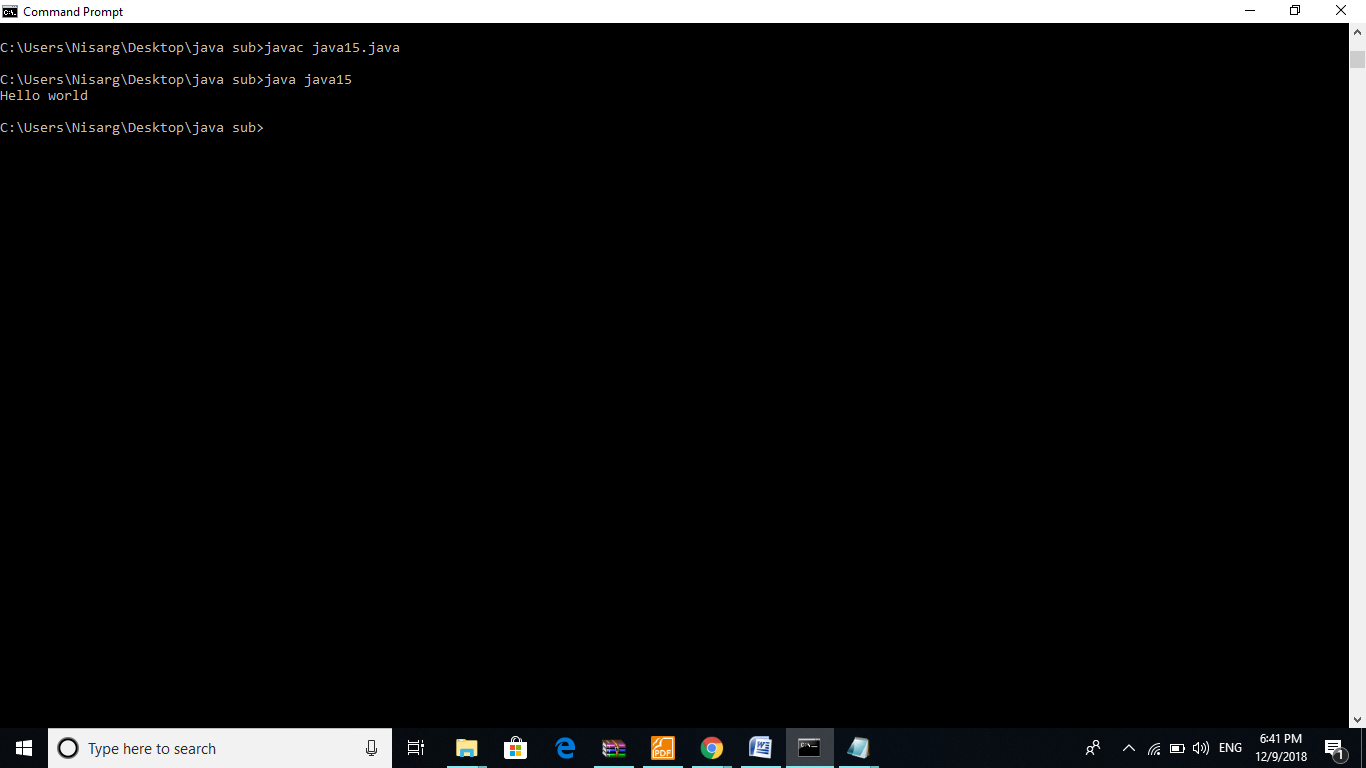
public static void main(String args[])

{

}

}

**Output:**



**16: Write programs in Java to use Wrapper class of each primitive data types**

**CODE:**

public class pro16

{

public static void main(String args[])

{

//Converting int primitive into Integer object

int num=100;

Character d = 'r';

Byte b = 13;

Byte a = 10;

Short s = 12;

Long l = 423545l;

Double e = 24.0;

Integer obj=Integer.valueOf(num);

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

System.out.println(d.charValue());

System.out.println(b.shortValue());

System.out.println(a.floatValue());

System.out.println(s.intValue());

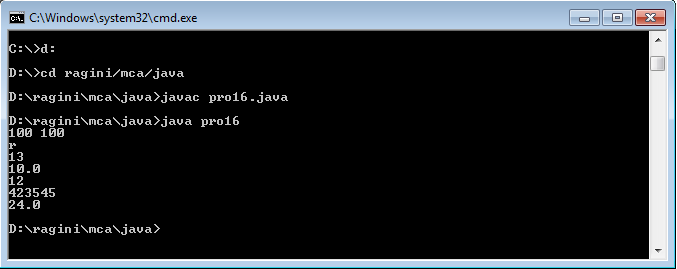
System.out.println(l.intValue() );

System.out.println(e.floatValue() );

}

}

**Output:**



**17. Write a class “circle” with radius as data member and count the number of instances created using default constructor only. [Constructor Chaining]**

**CODE:**

public class circle

{

static int count;

private double radius;

static

{

count=0;

}

circle()

{

count++;

}

circle(double rad)

{

this();

radius=rad;

}

public double getrad()

{

return radius;

}

public static void main(String[] args)

{

circle c=new circle(5);

System.out.println(c.getrad());

circle a=new circle(4);

System.out.println(a.getrad());

circle c1=new circle(3);

System.out.println(c1.getrad());

circle c2=new circle(2);

System.out.println(c2.getrad());

circle c3=new circle(1);

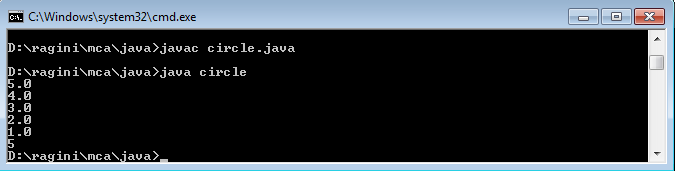
System.out.println(c3.getrad());

System.out.print(circle.count);

}

}

**Output:**



**18. Create a class Vehicle with data member vehicle\_type. Inherit the class in a class called car with data member model\_type, company name etc. display the information of the vehicle by defining the display function in both super and sub class [ Method Overriding]**

**CODE:**

**Vehicle.java**

public class vehicle

{

String vehicle\_type;

//Constructor

public vehicle(String type)

{

vehicle\_type=type;

}

//Methods

public String getvehicletype()

{

return vehicle\_type;

}

public void display()

{

System.out.println("Your vehicle type is :" + vehicle\_type);

}

}

**Car.java**

public class car extends vehicle

{

String model\_type,company\_name;

String veh\_type=super.vehicle\_type;

public car(String mtype,String companyname,String veh\_type)

{

super(veh\_type);

model\_type=mtype;

company\_name=companyname;

}

public void setdetails(String mtype,String cname)

{

model\_type=mtype;

company\_name=cname;

}

public void display()

{

System.out.println("\n Model Name : " + model\_type + "\n Company Name : "+ company\_name+"\nVehicle type : "+veh\_type);

}

}

**test18.java**

public class test18

{

public static void main(String args[])

{

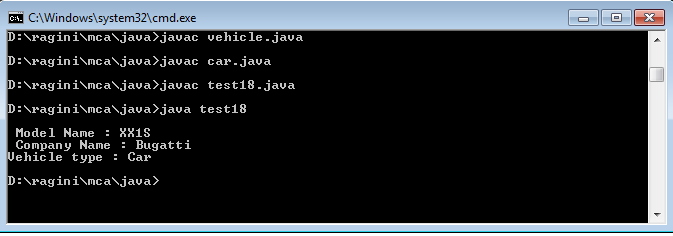
car c1=new car("XX1S","Bugatti","Car");

c1.display();

}

}

**Output:**



**19. Create a class “Account” containing accountNo, and balance as data members.**

**Derive the Account class into two classes named “Savings” and “Current”. The “Savings” class should contain a data member named interestRate, and the “Current” class should contain a data member called overdraftLimit. Create appropriate member functions for all the classes to enable functionalities to check balance, deposit, and withdraw amount in Savings and Current account.**

**[Ensure that the Account class cannot be instantiated.]**

**20. Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values. [ refer class Account and sub classes savingAccount and CurrentAccount in Q 19 for this task]**

**CODE:**

**Account.java**

public abstract class Account

{

protected long acc\_number;

protected static double accbalance=0.0;

protected String acc\_type=new String();

Account(long accno,String acc\_type)

{

acc\_number=accno;

}

double checkBalance()

{

return accbalance;

}

String deposit(double depamount)

{

accbalance=accbalance + depamount;

String accnt="Your amount is deposited and your current balance is "+accbalance;

return accnt;

}

void withdraw(double withdrawamt)

{

if(accbalance<100)

{

System.out.println(" Insufficient funds ");

}

else

{

accbalance=accbalance-withdrawamt;

}

}

}

**Current.java**

public class current extends Account

{

double overdraftLimit;

current(long accno,String acc\_type)

{

super(accno,acc\_type);

}

}

**Savings.java**

public class savings extends Account

{

protected double interestRate=6.00;

savings(long accno,String accounttype)

{

super(accno,accounttype);

acc\_type=accounttype;

}

public double add\_interest()

{

double Monthly\_Interest = super.accbalance\* (30) \* interestRate / (365);

return Monthly\_Interest;

}

}

**Main20.java**

class main20

{

public static void main(String args[])

{

savings sa=new savings(1,"withdraw");

current cu=new current(2,"savings");

System.out.println(sa.deposit(10000));

System.out.println(cu.deposit(20000));

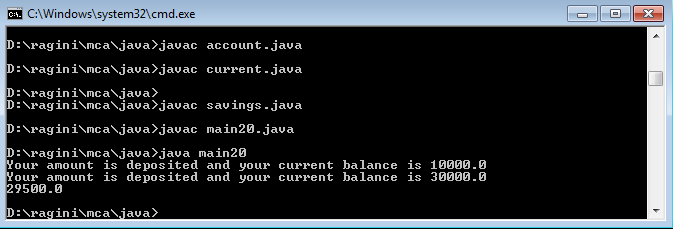
sa.withdraw(500.00);

System.out.println(sa.checkBalance());

}

}

**Output:**



**21. Write a program in Java to demonstrate the use of 'final' keyword in the field declaration. How it is accessed using the objects.**

**CODE:**

**twenty1.java**

import java.util.\*;

class abc

{

final int a=10;

int b=22;

void show()

{

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

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

}

void incre()

{

System.out.println("After increment:");

//++a; value of a cannot be changed

++b;

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

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

}

}

public class twenty1

{

public static void main(String args[])

{

abc aa=new abc();

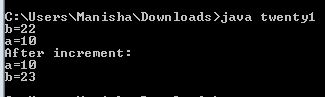
aa.show();

aa.incre();

}

}

**Output:**

****

**22. Write a java program to illustrates how to access a hidden variable. Class A declares a static variable x. The class B extends A and declares an instance variable x. display ( ) method in B displays both of these variables.**

**CODE:**

**twenty2.java**

import java.util.\*;

class A

{

public static int x=10;

}

class B extends A

{

int x=110;

int getx(){ return x; }

void display()

{

System.out.println("Static="+x);

System.out.println("Instance="+x);

}

}

public class twenty2

{

public static void main(String args[])

{

A a=new A();

System.out.println("Static="+a.x);

B b=new B();

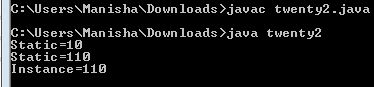
b.getx();

b.display();

}

}

**Output:**



**23. Describe abstract class called Shape which has three subclasses say Triangle,Rectangle, and Circle. Define one method area () in the abstract class and override this area () in these three subclasses to calculate for specific object i.e. area () of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle**

**CODE:**

**twenty3.java**

import java.util.\*;

abstract class shape

{

double area;

abstract public void area();

}

class triangle extends shape

{

double h,b;

void set(double h,double b)

{

this.h=h;

this.b=b;

area=0.5\*h\*b;

}

public void area()

{

System.out.println("Area of triangle is="+area);

}

}

class rectangle extends shape

{

double l,b;

void set(double l,double b)

{

this.l=l;

this.b=b;

area=2\*l\*b;

}

public void area()

{

System.out.println("Area of rectangle is="+area);

}

}

class circle extends shape

{

double r;

void set(double r)

{

this.r=r;

area=3.14\*r\*r;

}

public void area()

{

System.out.println("Area of circle is="+area);

}

}

public class twenty3

{

public static void main(String args[])

{

double height,breadth,l,b,r;

Scanner sc=new Scanner(System.in);

triangle t=new triangle();

System.out.println("Triangle:");

System.out.println("Enter height:");

height=sc.nextDouble();

System.out.println("Enter breadth:");

breadth=sc.nextDouble();

t.set(height,breadth);

t.area();

rectangle rect=new rectangle();

System.out.println("Rectangle:");

System.out.println("Enter lenght:");

l=sc.nextDouble();

System.out.println("Enter breadth:");

b=sc.nextDouble();

rect.set(l,b);

rect.area();

circle c=new circle();

System.out.println("circle:");

System.out.println("Enter radius:");

r=sc.nextDouble();

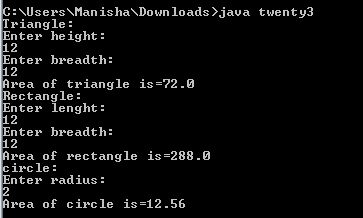
c.set(r);

c.area();

}

}

**Output:**

****

**24. Write a java program to implement an interface called Exam with a method Pass (int mark) that returns a boolean. Write another interface called Classify with a method Division (int average) which returns a String. Write a class called Result which implements both Exam and Classify. The Pass method should return true if the mark is greater than or equal to 50 else false. The Division method must return “First” when the parameter average is 60 or more, “Second” when average is 50 or more but below 60, “No division” when average is less than 50.**

**CODE:**

**twenty4.java**

import java.util.Scanner;

interface Exam

{

boolean Pass(int mark);

}

interface Classify

{

String Division(int average);

}

class Result implements Exam,Classify{

public boolean Pass(int mark)

{

if(mark>=50)

return true;

else

return false;

}

public String Division(int average)

{

if(average>=60)

return "First";

else if(average<60 && average>=50)

{

return "Second";

}

else

{

return "No Division";

}

}

}

public class twenty4

{

public static void main(String args[])

{

boolean p;

String d;

int marks,avg;

Scanner sc=new Scanner(System.in);

Result r=new Result();

System.out.println("Enter marks:");

marks=sc.nextInt();

System.out.println("Enter Average");

avg=sc.nextInt();

p=r.Pass(marks);

System.out.println("Pass="+p);

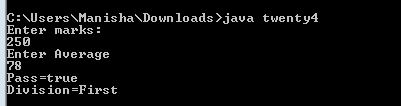
d=r.Division(avg);

System.out.println("Division="+d);

}

}

**Output:**

****

**25. Assume that there are two packages, student and exam. A student package contains Student class and the exam package contains Result class. Write a program that generates mark sheet for students.**

**CODE:**

**PL25.java**

import java.util.Scanner;

import student.Student;

import exam.Exam;

public class PL25

{

public static void main(String a[])

{

Student s = new Student();

Exam e = new Exam();

//s.set(3001,"Jim");

//System.out.println("Enrol: " + s.getEnrol());

int enrol, marks;

String name, result;

float per;

Scanner sc = new Scanner(System.in);

System.out.println("Enter enrol no: ");

enrol=sc.nextInt();

System.out.println("Enter name: ");

name=sc.next();

System.out.println("Enter total marks: ");

marks=sc.nextInt();

System.out.println("Enter percentage: ");

per=sc.nextFloat();

System.out.println("Enter result (class): ");

result=sc.next();

s.set(enrol,name);

e.set(marks,per,result);

System.out.println("=================================================”);

System.out.println(" MARKSHEET ");

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

System.out.println("Enrolment No : " + s.getEnrol());

System.out.println("Student\'s Name : " + s.getName());

System.out.println("Total Marks : " + e.getMarks());

System.out.println("Percentage : " + e.getPer());

System.out.println("Result (class) : " + e.getResult());

}

}

**Exam.java**

package exam;

public class Exam

{

int marks;

float per;

String result;

public void set(int marks, float percent, String res)

{

this.marks=marks;

per=percent;

result=res;

}

public int getMarks()

{

return marks;

}

public float getPer()

{

return per;

}

public String getResult()

{

return result;

}

}

**Student.java**

/\* Assume that there are two packages, student and exam. A student package contains Student class

and the exam package contains Result class. Write a program that generates mark sheet for students. \*/

package student;

public class Student

{

int eno;

String name;

public void set(int num, String nm)

{

eno=num;

name=nm;

}

public int getEnrol()

{

return eno;

}

public String getName()

{

return name;

}

}

**Output:**

**26. Define a class A in package apack. In class A, three variables are defined of access modifiers protected, private and public. Define class B in package bpack which extends A and write display method which accesses variables of class A. Define class C in package cpack which has one method display() in that create one object of class A and display its variables. Define class ProtectedDemo in package dpack in which write main () method. Create objects of class B and C and class display method for both these objects.**

**CODE:**

**apack.java**

package apack;

public class A

{

private int a1=10;

public int a2=20;

protected int a3=30;

}

**bpack.java**

package bpack;

import apack.\*;

public class B extends A

{

public void display()

{

//System.out.println(a1); //a1 is private so can't be used

System.out.println(a2);

System.out.println(a3);

}

}

**cpack.java**

package cpack;

import apack.\*;

public class C

{

public void display()

{

A a11=new A();

System.out.println(a11.a2);

}

}

**dpack.java:**

package dpack;

import bpack.\*;

import cpack.\*;

public class D

{

public static void main (String[] args)

{

B b1=new B();

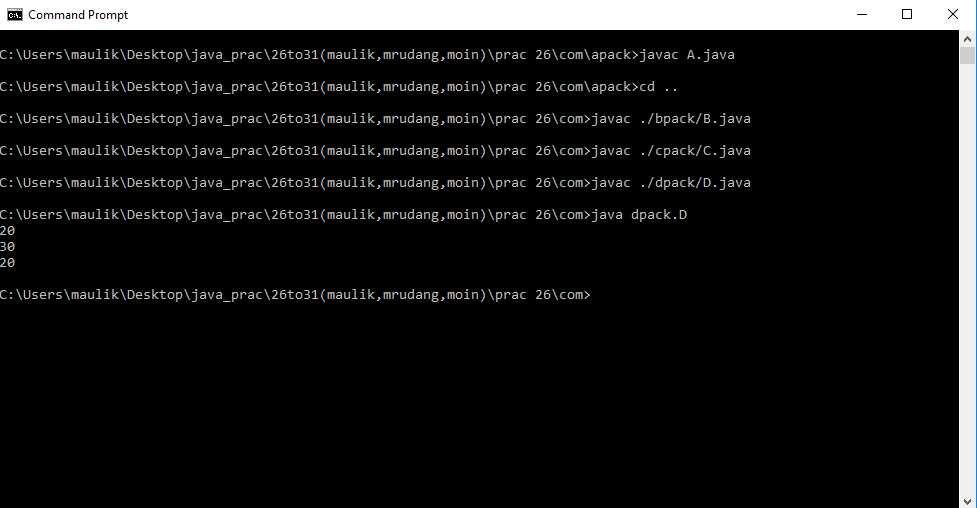
b1.display();

C c1=new C();

c1.display();

}

}

**OUTPUT:**

**27. Write a java program to implement lambda expression with functional interface in java:**

**CODE:**

@FunctionalInterface

interface TestInterface

{

public abstract void show(int n);

public static void hello()

{

System.out.println("static method");

}

public default void test()

{

System.out.println("default method");

}

}

class pro27

{

public static void main(String[] args)

{

int x=10;

TestInterface.hello();

TestInterface ti=(n)->

{

System.out.println(n);

};

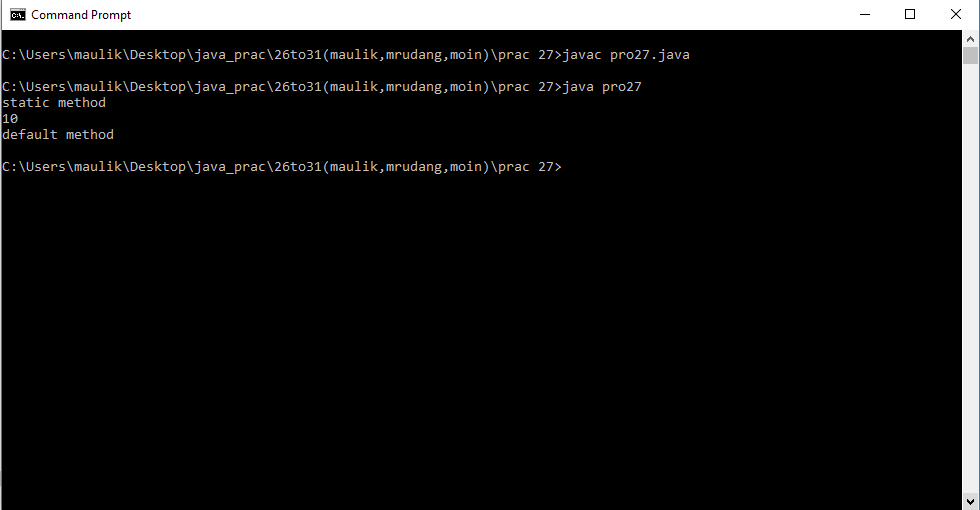
ti.show(x);

ti.test();

}

}

**OUTPUT:**

****

**28. Write a java program to accept string check whether it is in Upper or Lower case. As per case change it in according vise versa:**

**CODE:**

import java.util.Scanner;

public class prac28

{

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

String S=scan.nextLine();

StringBuilder sb=new StringBuilder(S);

char c;

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

{

c=S.charAt(i);

if(Character.isLowerCase(c))

{

sb.setCharAt(i,Character.toUpperCase(c));

}

if(Character.isUpperCase(c))

{

sb.setCharAt(i,Character.toLowerCase(c));

}

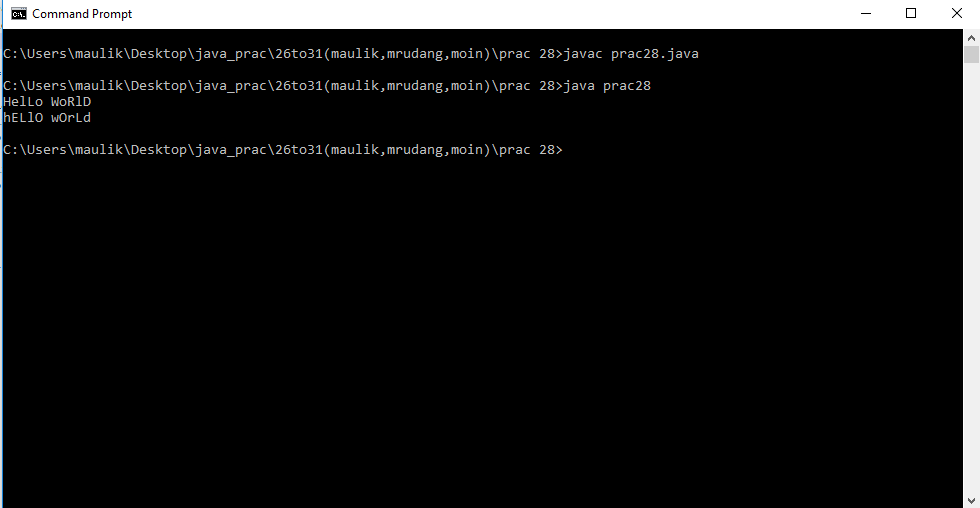
}

System.out.println(sb.toString());

}

}

**OUTPUT:**

****

**29. Write a java program to use important methods of String class.**

**CODE:**

public class prac29

{

public static void main(String[] args)

{

String s=" hello world.. ";

String s1="Greetings!!";

System.out.println(" substring " + s.substring(0,3));

System.out.println(" string length " + s.length());

System.out.println(" replace " + s.replace(s,s1));

System.out.println(" concate " + s.concat(s1));

System.out.println(" contains " + s.contains("world"));

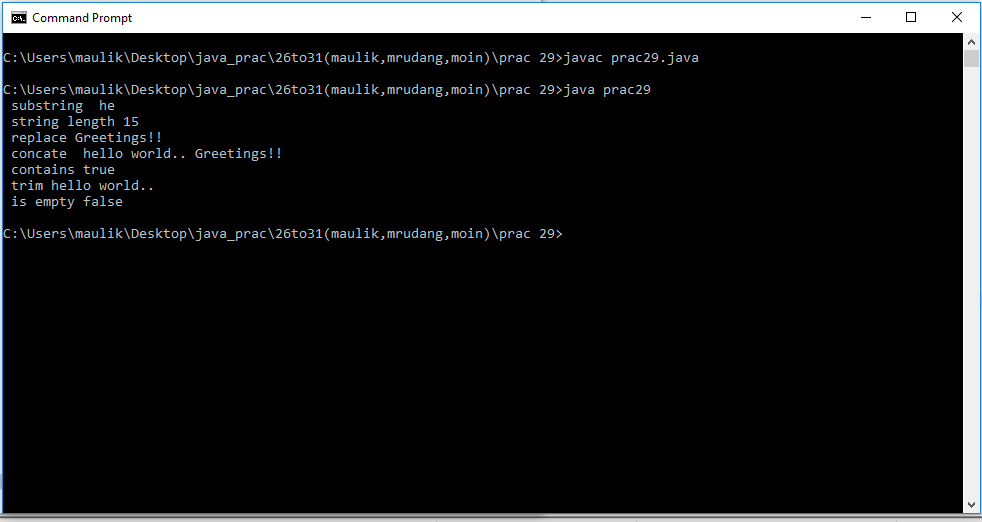
System.out.println(" trim " + s.trim());

System.out.println(" is empty " + s.isEmpty());

}

}

**OUTPUT:**

****

**30. Write a program in Java to demonstrate use of final class**.

**CODE:**

final class Bike{}

class Honda1 extends Bike

{

void run(){System.out.println("running safely with 100kmph");

}

public static void main(String args[])

{

Honda1 honda= new Honda1();

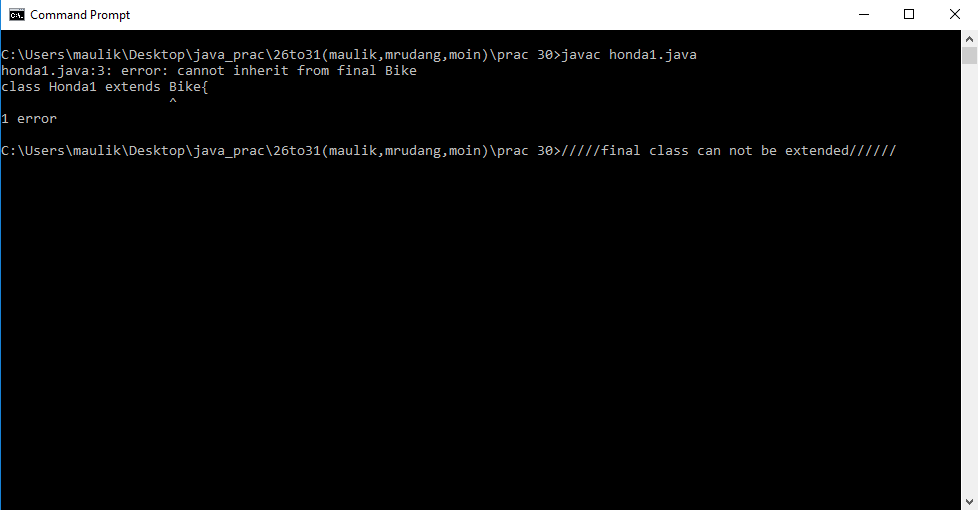
honda.run();

}

}

//final class cannot inherited by any other class. so it will generate error**.**

**OUTPUT:**

****

**31. Write a program in Java to develop user defined exception for 'Divide by Zero' error:**

**CODE:**

class dividebyzero extends Exception

{

public dividebyzero(String s)

{

super(s);

}

}

public class prac31

{

void check(int a,int b) throws dividebyzero

{

if(b==0)

{

throw new dividebyzero("can not divide by zero");

}

}

public static void main(String[] args)

{

int a=10;

prac31 p=new prac31();

try

{

p.check(10,0);

}

catch(dividebyzero obj)

{

System.out.println("try something else");

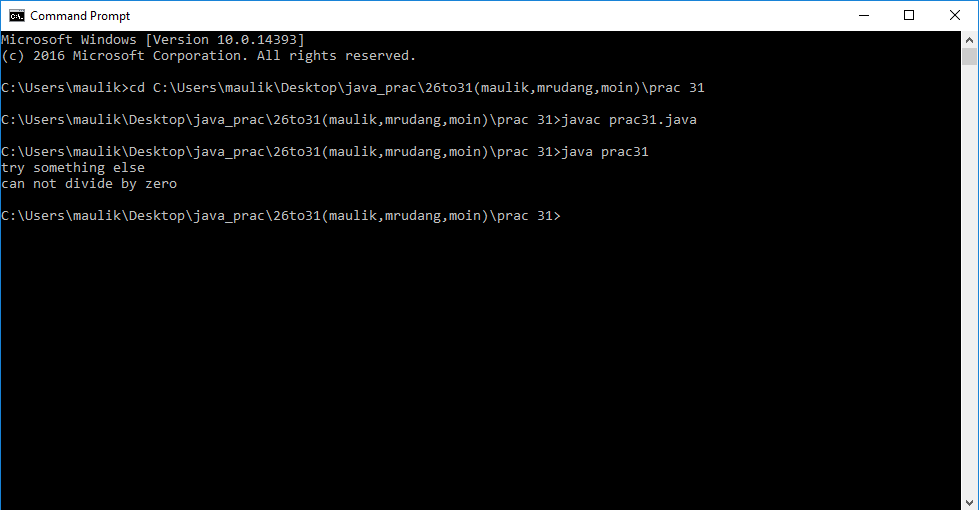
System.out.println(obj.getMessage());

}

}

}

**OUTPUT:**

****

**32. Write a program in Java to demonstrate throw, throws, finally, multiple try block and multiple catch exception.**

**CODE:**

import java.util.\*;

//Exception Handling: 32

class ExceptionDemo

{

void myMethod(int num) throws ArithmeticException

{

if(num==0)

{

throw new ArithmeticException();

}

else

{

int ans = 50 / num;

System.out.println("\n ans:" + ans);

}

}

public static void main(String args[])

{

int no=0;

Scanner sc = new Scanner(System.in);

try

{

String name;

System.out.print("\nEnter the value:");

name = sc.next();

no =Integer.parseInt(name);

try

{

int ans=50/no;

}

catch(ArithmeticException e)

{

System.out.println("NO ZERO ALLOWED!!!");

}

}

catch(NumberFormatException ne)

{

System.out.println(ne.getMessage());

System.out.println("Input only number !!!");

}

catch(Exception ne)

{

System.out.println(ne.getMessage());

System.out.println("An Error has occured !!!");

}

finally

{

System.out.print("\nEnter the value again:");

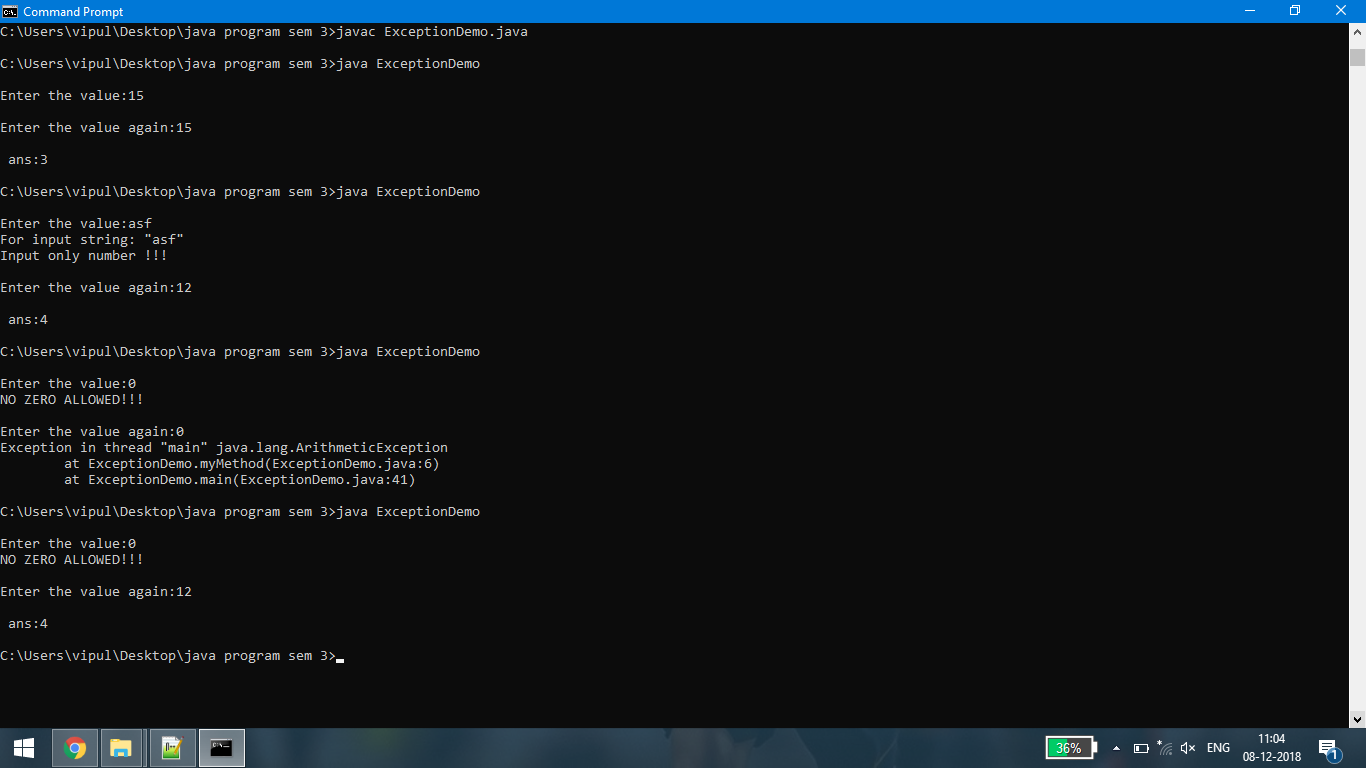
no = sc.nextInt();

ExceptionDemo mobj = new ExceptionDemo();

mobj.myMethod(no);

}

}

**Output:**

**33. Write a small application in Java to develop Banking Application in which user deposits the amount Rs 1000.00 and then start withdrawing of Rs 400.00, Rs 300.00 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 500 thereafter**

**CODE:**

import java.util.Scanner;

class CustomException extends Exception

{

CustomException(String msg)

{

super(msg);

}

}

class Bank

{

double bal=0;

void deposite(double amt)

{

bal = bal + amt;

}

double withdraw(double amt)

{

try

{

if(amt > bal)

{

amt=0;

throw new CustomException("Not Sufficient Fund");

}

else

{

bal = bal - amt;

}

}

catch(CustomException e)

{

System.out.println(e.getMessage());

}

finally

{

return amt;

}

}

}

class UserDefinedException

{

public static void main(String args[])

{

double amt;

Scanner sc = new Scanner(System.in);

Bank b = new Bank();

System.out.println("Enter the amount do you want deposite:");

amt = sc.nextDouble();

b.deposite(amt);

do

{

System.out.println("Enter the amount do you want withdraw:");

amt = sc.nextDouble();

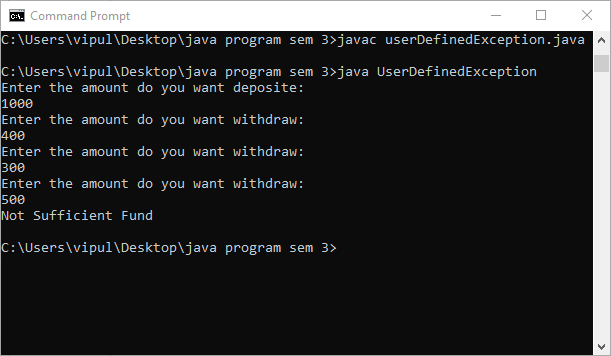
amt=b.withdraw(amt);

}while(amt!=0);

}

}

**Output:**



**34. Write a program to write at least 10 objects of the Circle class in a File using ObjectOutputStream and perform basic operations: adding, retrieving, updating, removing elements. [ Use Generic Data types and Collections for the this task]**

**CODE:**

import java.io.\*;

import java.util.Scanner;

public class Circle implements Serializable

{

private int radius;

public int area;

public void calculateArea()

{

area = 22/7 \* radius \* radius;

}

int getRadius()

{

return radius;

}

void setRadius(int r)

{

radius = r;

}

public static void main(String [] args)

{

Scanner scanner = new Scanner(System.in);

Circle c[] = new Circle[10];

int r=0;

char ch;

String str="";

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

c[i] = new Circle();

System.out.println("Enter Radius: ");

r = scanner.nextInt();

c[i].setRadius(r);

c[i].calculateArea();

System.out.println("Area for this circle is : "+c[i].area);

}

try

{

FileOutputStream fileOut = new FileOutputStream("./Circle.ser");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

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

{

out.writeObject(c[i]);

}

out.close();

fileOut.close();

System.out.println("Serialized data is saved in ./Circle.ser");

}

catch (IOException i)

{

i.printStackTrace();

}

try

{

FileInputStream fileIn = new FileInputStream("./Circle.ser");

ObjectInputStream in = new ObjectInputStream(fileIn);

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

{

c[i] = (Circle) in.readObject();

System.out.println("Deserialized Circle "+i+1+" : ");

System.out.println("Radius: " + c[i].radius);

System.out.println("Area: " + c[i].area);

System.out.println("for edit enter 'e', for delete enter 'r', another to

continue with list");

ch = scanner.next().charAt(0);

if(ch=='e')

{

System.out.println("Enter new radius: ");

r = scanner.nextInt();

c[i].setRadius(r);

c[i].calculateArea();

System.out.println("Area for this circle is : "+c[i].area);

str=str+Integer.toString(0);

}

else if(ch=='r')

{

str=str+Integer.toString(1);

System.out.println("This object will be deleted");

}

else

{

str=str+Integer.toString(0);

}

}

in.close();

fileIn.close();

//System.out.println(str);

try

{

FileOutputStream fileOut =

new FileOutputStream("./Circle.ser");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

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

{

if(str.charAt(i)!='0')

{

System.out.println("Radius: " + c[i].radius+ " is deleted...");

}

else

{

out.writeObject(c[i]);

}

}

out.close();

fileOut.close();

System.out.println("Serialized data is saved in ./Circle.ser");

}

catch (IOException i)

{

i.printStackTrace();

}

}

catch (IOException i)

{

i.printStackTrace();

return;

}

catch (ClassNotFoundException ec)

{

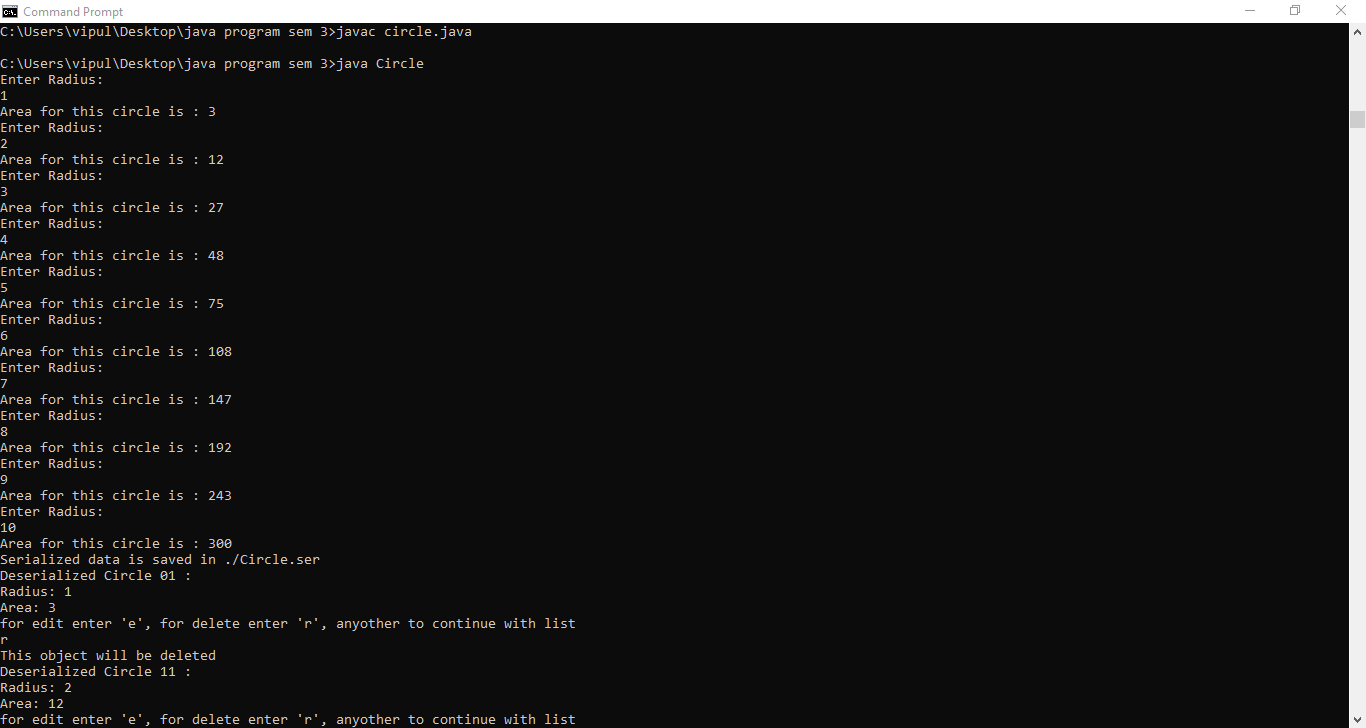
System.out.println("Circle class not found");

ec.printStackTrace();

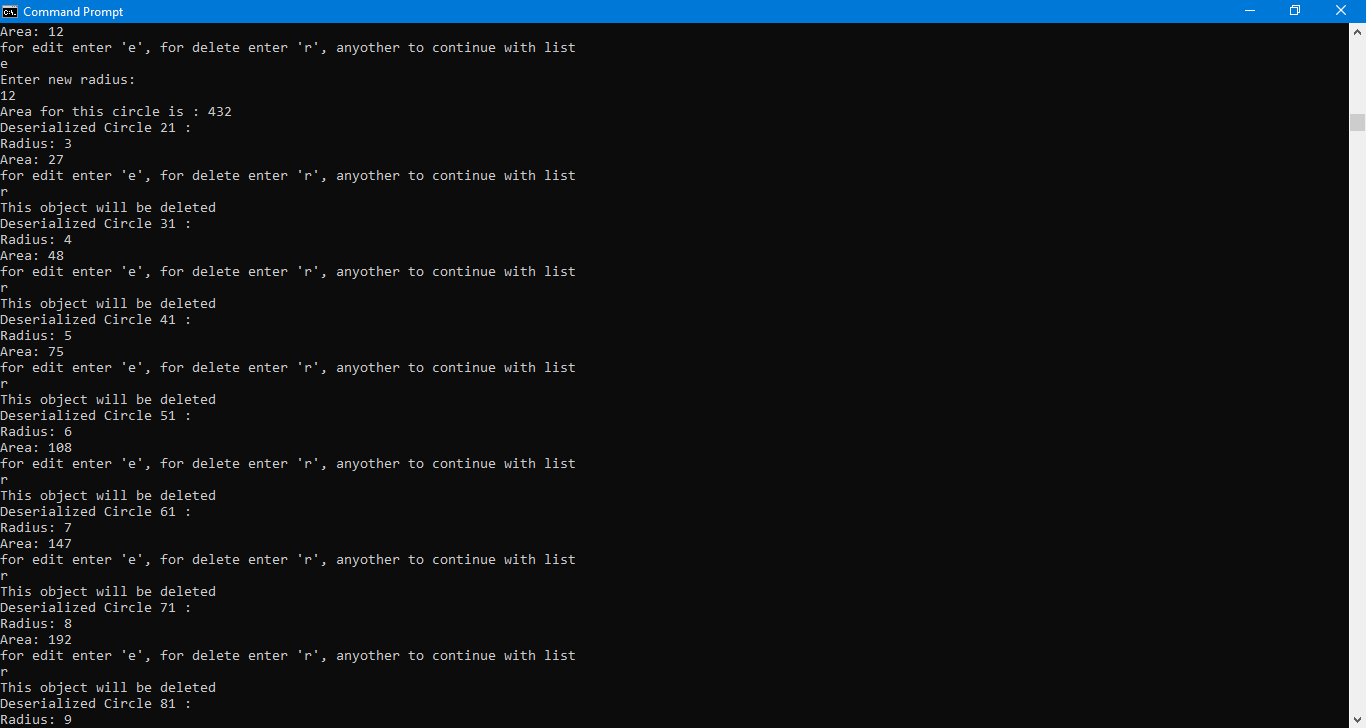
return;

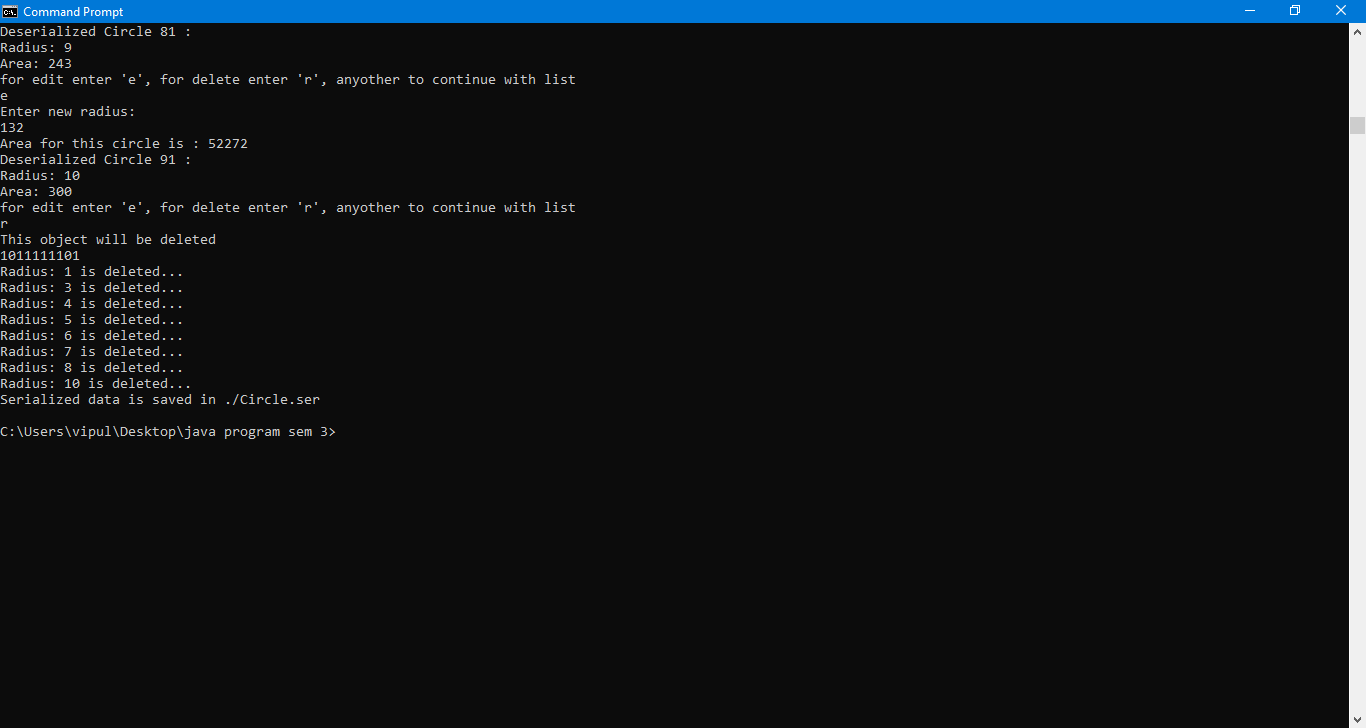
}

}

}

**Output:**





**35. Write a program for Java Generics and Collections Sorting operations: 1. Sorting a list according to natural ordering of elements 2. Reversing sort order 3. Sorting a list whose elements of a custom type 4. Sorting a list using a Comparator**

**CODE:**

**Employee.java:-**

public class Employee implements Comparable<Employee>

{

private String name;

private int age;

private int salary;

public Employee(String name, int age, int salary)

{

this.name = name;

this.age = age;

this.salary = salary;

}

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

public int getAge()

{

return age;

}

public void setAge(int age)

{

this.age = age;

}

public int getSalary()

{

return salary;

}

public void setSalary(int salary)

{

this.salary = salary;

}

public String toString()

{

return String.format("(%s, %d, %d)", name, age, salary);

}

**@Override**

public int compareTo(Employee employee)

{

return employee.salary - this.salary;

}

}

**EmployeeAgeComparator.java:-**

import java.util.Comparator;

public class EmployeeAgeComparator implements Comparator<Employee>

{

**@Override**

public int compare(Employee emp1, Employee emp2)

{

return emp1.getAge() - emp2.getAge();

}

}

**JavaListSortingExamples.java:-**

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.Date;

import java.util.List;

public class JavaListSortingExamples

{

public static void main(String[] args)

{

System.out.println("===== SORTING A LIST OF STRINGS =====");

List<String> listStrings = Arrays.asList("Orange", "Grape", "Apple", "Lemon",

"Banana");

System.out.println("Before sorting: " + listStrings);

Collections.sort(listStrings);

System.out.println("After sorting: " + listStrings);

System.out.println("\n===== SORTING A LIST OF CHARACTERS =====");

List<Character> listCharacters = Arrays.asList('F', 'C', 'A', 'B', 'Z', 'E', 'K', 'P');

System.out.println("Before sorting: " + listCharacters);

Collections.sort(listCharacters);

System.out.println("After sorting: " + listCharacters);

System.out.println("\n===== SORTING A LIST OF INTEGERS =====");

List<Integer> listIntegers = Arrays.asList(1, 6, 9, 3, 2, 0, 8, 4, 7, 5);

System.out.println("Before sorting: " + listIntegers);

Collections.sort(listIntegers);

System.out.println("After sorting: " + listIntegers);

System.out.println("\n===== SORTING A LIST OF DATES =====");

List<Date> listDates = new ArrayList<Date>();

DateFormat dateFormatter = new SimpleDateFormat("yyyy-MM-dd");

try

{

listDates.add(dateFormatter.parse("2013-09-30"));

listDates.add(dateFormatter.parse("2013-07-06"));

listDates.add(dateFormatter.parse("2013-11-28"));

}

catch (ParseException ex)

{

System.err.print(ex);

}

System.out.println("Before sorting: " + listDates);

Collections.sort(listDates);

System.out.println("After sorting: " + listDates);

System.out.println("\n===== REVERSING ORDER =====");

System.out.println("Before sorting: " + listIntegers);

Collections.sort(listIntegers);

System.out.println("After sorting: " + listIntegers);

Collections.reverse(listIntegers);

System.out.println("After reversing: " + listIntegers);

System.out.println("\n===== SORTING A LIST OF CUSTOM TYPES =====");

List<Employee> listEmployees = new ArrayList<Employee>();

listEmployees.add(new Employee("Tom", 45, 80000));

listEmployees.add(new Employee("Sam", 56, 75000));

listEmployees.add(new Employee("Alex", 30, 120000));

listEmployees.add(new Employee("Peter", 25, 60000));

System.out.println("Before sorting: " + listEmployees);

Collections.sort(listEmployees);

System.out.println("After sorting: " + listEmployees);

System.out.println("\n===== SORTING A LIST USING A COMPARATOR

=====");

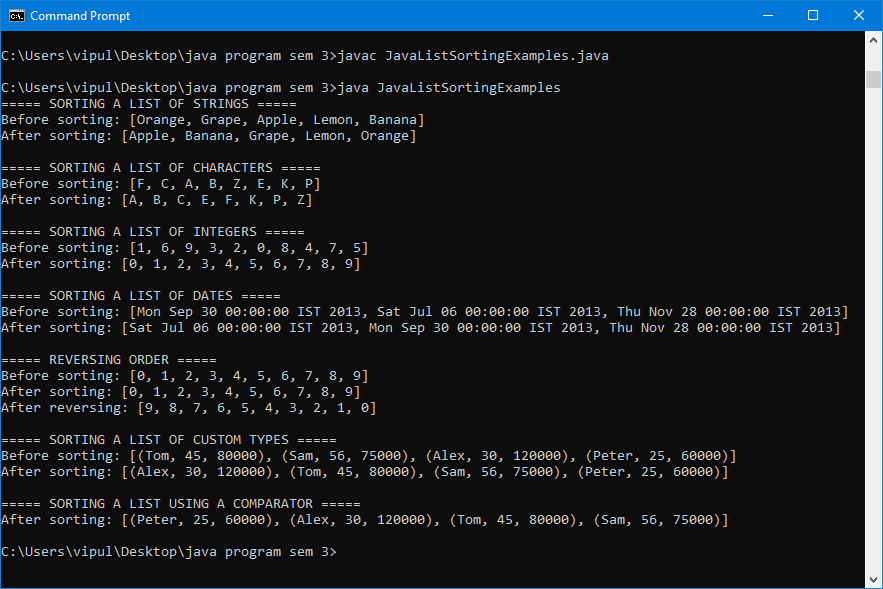
Collections.sort(listEmployees, new EmployeeAgeComparator());

System.out.println("After sorting: " + listEmployees);

}

}

**Output:**



**36. Write a program in Java to create, write, modify, read operations on a Text file**

**CODE:**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class EditFile

{

public static void main(String[] args)

{

try

{

String verify, putData;

File file = new File("file.txt");

file.createNewFile();

FileWriter fw = new FileWriter(file);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("Some text here for a reason");

bw.flush();

bw.close();

FileReader fr = new FileReader(file);

BufferedReader br = new BufferedReader(fr);

verify=br.readLine();

System.out.println("File Content before: " + verify);

br.close();

fw = new FileWriter(file);

bw = new BufferedWriter(fw);

if(verify != null)

{

putData = verify.replaceAll("here", "there");

System.out.println("File Content after change: " + putData);

bw.write(putData);

bw.flush();

bw.close();

}

}

catch(IOException e)

{

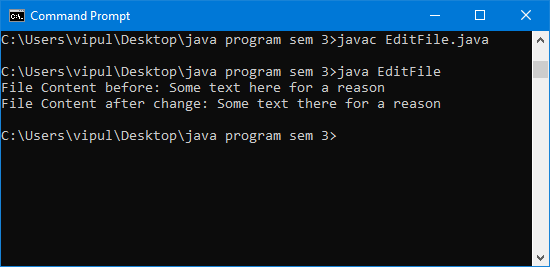
e.printStackTrace();

}

}

}

**Output:**



**37 Write a java program to illustrates use of standard input stream to read the user input.**

**CODE:**

// UserInput.java

import java.util.Scanner;

class UserInput

{

public static void main(String[] args)

{

// Create a scanner to wrap the standard input stream

Scanner scanner = new Scanner(System.in);

// Prompt user to enter a string and press enter

System.out.print("Enter your name: ");

System.out.println("Hello, " + scanner.nextLine());

// Use the scanner to then get a numeric value from the user.

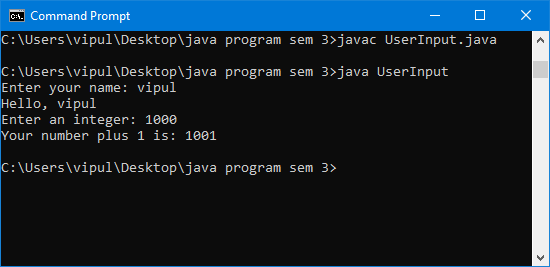
System.out.print("Enter an integer: ");

System.out.println("Your number plus 1 is: " + (scanner.nextInt() + 1));

}

}

**Output:**



**38. Write a Java program to checks the existence of a specified file.**

**CODE:**

import java.io.File;

import java.io.IOException;

public class Prog38

{

public static void main(String args[])

{

File temp;

// File temp1=new File("hello","txt");

try

{

temp=File.createTempFile("myTempFile","txt");

boolean exists=temp.exists(); //true

//boolean exists=temp1.exists(); false

System.out.println("Temp file exists: "+exists);

}

catch(IOException e)

{

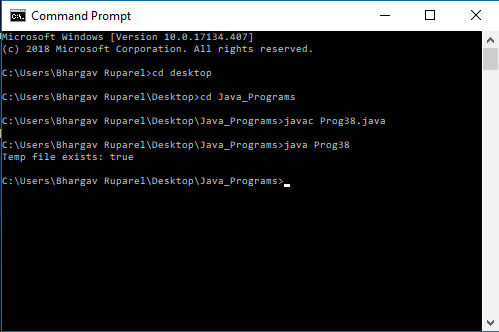
e.printStackTrace();

}

}

}

**Output :-**



**39. Write a Java program to create a file to the specified location.**

**CODE:**

import java.io.\*;

public class Prog39

{

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

{

File f1 = new File("MyFolder");

File f2 = new File("MyFolder/test.txt");

f1.mkdir();

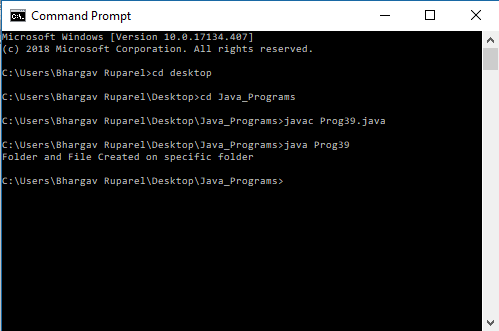
f2.createNewFile();

System.out.println("Folder and File Created on specific folder");

}

}

**Output:**



**40. Write a java program to demonstrate the way contents are read from a file.**

**CODE:**

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

/\*\*

\* Example class for FileReader

\*/

public class Prog40

{

//utility method to print a char

static void print(char value)

{

System.out.print(value);

}

public static void main(String[] args)

{

readFile();

}

static void readFile()

{

try

{

//create a FileReader Object by providing File name in the constructor

FileReader reader = new

FileReader("D:/Yash/LD/JAVA/PracticalPrograme/39/MyFolder/test.txt");

int c; //this will contain the character value as int

while ((c = reader.read()) != -1)

{

print((char) c);

}

//close the reader after reading is completed

reader.close();

}

catch (FileNotFoundException e)

{

e.printStackTrace();

}

catch (IOException e1)

{

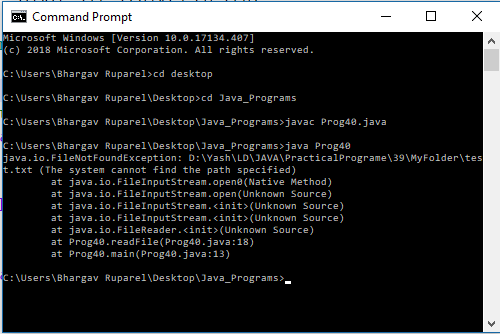
e1.printStackTrace();

}

}

}

**Output :-**



**41. Write a java program to first checks the existence of a specified file .If the file exists , the data is written to the file through the object of the FileOutputStream class .**

**CODE:**

import java.io.File;

import java.io.FileOutputStream;

public class Prog41

{

public static void main(String[] args)

{

try

{

File f = new File("test.txt");

if(f.exists() && !f.isDirectory())

{

System.out.println("File Exists");

/\*Write\*/

FileOutputStream fout=new FileOutputStream("test.txt");

fout.write(65);

String s="Write with fileoutputstream .";

byte b[]=s.getBytes();//converting string into byte array

fout.write(b);

fout.close();

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

}

else

{

System.out.println("File Does Not Exists");

}

}

catch(Exception e)

{

// if any error occurs

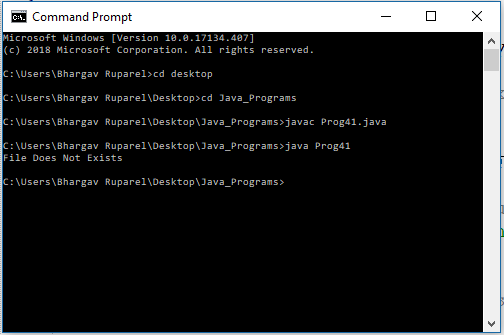
e.printStackTrace();

}

}

}

**Output :-**



**42. Write a java program to count the availability of text lines in the particular file . A file is read before counting lines of a particular file .**

**CODE:**

import java.io.\*;

public class Prog42

{

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

{

File file = new File("test.txt");

/\*read\*/

BufferedReader br = new BufferedReader(new FileReader(file));

String st;

while ((st = br.readLine()) != null)

System.out.println(st);

FileInputStream fileStream = new FileInputStream(file);

InputStreamReader input = new InputStreamReader(fileStream);

BufferedReader reader = new BufferedReader(input);

String line;

// Initializing counters

int sentenceCount = 0;

// Reading line by line from the

// file until a null is returned

while((line = reader.readLine()) != null)

{

if(!(line.equals("")))

{

// [!?.:]+ is the sentence delimiter in java

String[] sentenceList = line.split("[!?.:]+");

sentenceCount += sentenceList.length;

}

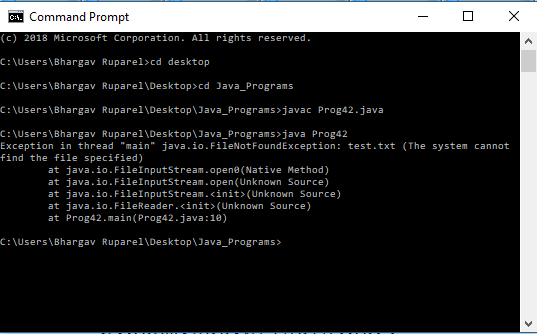
}

System.out.println("Total number of sentences = " + sentenceCount);

}

}

**Output:**



**43. Write a generic method to count the number of elements in a collection that have a specific property ( for example , odd integers , prime numbers , palindromes ).**

**CODE:**

import java.util.List;

import java.util.Arrays;

import java.io.\*;

public class Test

{

public interface SomeSpecialNumber<T>

{

public boolean match(T t);

}

public class PrimeNumber implements SomeSpecialNumber<Integer>

{

@Override

public boolean match(Integer t)

{

for (int i = 2; i < t; ++i)

{

if (t % i == 0)

{

return false;

}

}

return true;

}

}

public class EvenNumber implements SomeSpecialNumber<Integer>

{

@Override

public boolean match(Integer t)

{

return (t % 2 == 0);

}

}

public class Algorithm

{

public <T> int count(List<T> list, SomeSpecialNumber<T> s)

{

int count = 0;

for (T type : list)

{

if (s.match(type))

{

++count;

}

}

return count;

}

}

public static void main(String[] args)

{

Test tq= new Test();

tq.testSpecificElement();

}

public void testSpecificElement()

{

List<Integer> lst = Arrays.asList(1, 4, 5, 7, 8, 9, 13, 11,12);

PrimeNumber pn = new PrimeNumber();

EvenNumber en = new EvenNumber();

Algorithm a = new Algorithm();

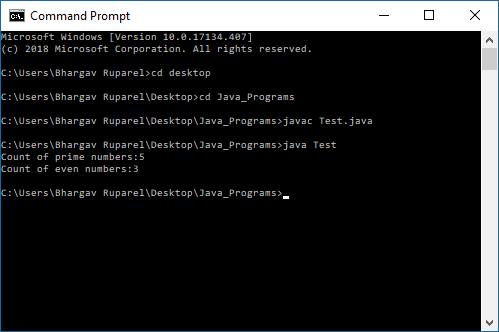
System.out.println("Count of prime numbers:" + a.count(lst, pn));

System.out.println("Count of even numbers:" + a.count(lst, en));

}

}

**Output:**



**44. Write a generic method to exchange the positions of two different elements in an array.**

**CODE:**

public final class Q44

{

public static <T> void swap(T[] a, int i, int j)

{

T temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

**45. Write a generic method to find the maximal element in the range [begin, end) of a list.**

**CODE:**

import java.util.\*;

public final class Q45

{

public static <T extends Object & Comparable<? super T>>

T max(List<? extends T> list, int begin, int end)

{

T maxElem = list.get(begin);

for (++begin; begin < end; ++begin)

if (maxElem.compareTo(list.get(begin)) < 0)

maxElem = list.get(begin);

return maxElem;

}

}

**46. Write a program to implement JDBC/ODBC connectivity to data base using java program.**

**CODE:**

import java.sql.\*;

public class Q46

{

public static void main(String args[])

{

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/java\_test","

root","root");

System.out.println("JDBC Connection Succefully Built");

con.close();

}

catch(Exception e)

{

System.out.println("Failed to connect the Database");

System.out.println(e);

}

}

}

**47. Write a java program to connect any of database e.g. MYSQL / Oracle /MS Access/ etc.**

**CODE:**

import java.sql.\*;

public class Q47

{

public static void main(String args[])

{

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/java\_test","root","root");

System.out.println("MYSQL Successfully Connected");

con.close();

}

catch(Exception e)

{

System.out.println("Failed to connect MYSQL Database");

System.out.println(e);

}

}

}

**48. Write a java program to create Employee table(Empno., Name, Designation, Salary) and insert a record in it.**

**CODE:**

import java.sql.\*;

public class Q48

{

public static void main(String args[])

{

Connection con=null;

Statement stmt=null;

try

{

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/java\_test","root","root");

stmt = conn.createStatement();

String sql = "CREATE TABLE Employee " +

"(Emp\_No INTEGER not NULL, " +

" Name VARCHAR(255), " +

" Designation VARCHAR(255), " +

" Salary INTEGER, " +

" PRIMARY KEY ( Emp\_No ))";

con.close();

stmt.executeUpdate(sql);

System.out.println("Created table in the database...");

sql = "INSERT INTO Employee " +

"VALUES(101, 'Sumit', 'Manager', 280000)";

stmt.executeUpdate(sql);

System.out.println("Successfully Inserted record into the table...");

}

catch(Exception e)

{

System.out.println("Failed to connect the Database");

System.out.println(e);

}

}

}

**49. Write a JAVA program to accept the details of student (Rno , SName , Per) from the user and insert it into the table. (use PreparedStatement Class).**

**CODE:**

import java.sql.\*;

public class Q49

{

public static void main(String args[])

{

Connection con=null;

Statement stmt=null;

Scanner sc=new Scanner(System.in);

int per,roll\_no;

String name;

try

{

Class.forName("com.mysql.jdbc.Driver"); con=DriverManager.getConnection("jdbc:mysql://localhost:3306/java\_test","root","root");

stmt = conn.createStatement();

String sql = "CREATE TABLE Student " +

"(R\_No INTEGER not NULL, " +

" SName VARCHAR(255), " +

" Per DECIMAL(5,2))";

con.close();

stmt.executeUpdate(sql);

System.out.println("Enter the Student's Roll No");

roll\_no=sc.nextInt();

System.out.println("Enter the Student's Name");

name=sc.next();

System.out.println("Enter the Student's Percentage");

per=sc.nextInt();

PreparedStatement st=con.prepareStatement("insert into

Student(R\_No,SName,Per) values(?,?,?)");

st.setInt(1,roll\_no);

st.setString(2,name);

st.setInt(3,per);

st.executeUpdate();

System.out.println("Successfully Inserted record into the table...");

}

catch(Exception e)

{

System.out.println("Failed to connect the Database");

System.out.println(e);

}

}

}

**50. Write a Menu driven program in Java for the following**.

**1. Create a Library Table (BookID, ISSNNo., Author, BookTitle, Price, Publisher, Year)**

**2. Insert Record into the Library Table.**

**3. Update The Existing Record.**

**4. Display all the Records from the Table.**

**5. Display names of books stating with “J” character**

**6. Delete the record**

**7. Exit from the program.**

**CODE:**

import java.sql.\*;

public class Q50

{

public static void main(String args[])

{

Connection con=null;

Statement stmt=null;

Scanner sc=new Scanner(System.in);

try

{

Class.forName("com.mysql.jdbc.Driver"); con=DriverManager.getConnection("jdbc:mysql://localhost:3306/java\_test","root","root");

stmt = conn.createStatement();

String sql = "CREATE TABLE Library " +

"(Book\_ID INTEGER not NULL, " +

" ISSN\_No VARCHAR(255), " +

" Author VARCHAR(255), " +

" Title VARCHAR(255), " +

" Price INTEGER," +

" Publisher VARCHAR(255), " +

" Year INTEGER," +

" PRIMARY KEY ( Book\_ID ))";

stmt.executeUpdate(sql);

int choice=1;

while(choice!=0)

{

System.out.println("1.Insert a Record");

System.out.println("2.Update a Record");

System.out.println("3.Display All Records");

System.out.println("4.Delete a Record");

System.out.println("5.Show Records

where Books Title Starts with J ");

System.out.println("0.Exit");

choice=sc.nextInt();

if(choice==1)

{

System.out.println("Enter the Book ID");

int id=sc.nextInt();

System.out.println("Enter the Book's ISSN No");

String issn=sc.next();

System.out.println("Enter the Title");

String title=sc.next();

System.out.println("Enter the Author Name");

String name=sc.next();

System.out.println("Enter the Price");

int price=sc.nextInt();

System.out.println("Enter the Publisher");

String publisher=sc.next();

System.out.println("Enter the Year");

int year=sc.nextInt();

PreparedStatement st=con.prepareStatement("insert into Library(Book\_ID,ISSN\_No,Author,Title,Price,Publisher,Year) values(?,?,?,?,?,?,?)");

st.setInt(1,id);

st.setString(2,issn);

st.setString(3,name);

st.setString(4,title);

st.setInt(5,price);

st.setString(6,publisher);

st.setInt(7,year);

st.executeUpdate();

System.out.println("Successfully Inserted record into the table...");

}

if(choice==2)

{

System.out.println("Enter the Id of the book to be updated");

int id =sc.nextInt();

System.out.println("Enter the new name of the book");

String name=sc.next();

String sql = "UPDATE Library " +

"SET Title = \'"+name+" WHERE Book\_ID ="+id+"";

stmt.executeUpdate(sql);

System.out.println("Successfully Updated Record");

}

if(choice==3)

{

String sql = "SELECT \* from Library";

ResultSet rs = stmt.executeQuery(sql);

while(rs.next())

{ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.print("ID: " + rs.getInt("Book\_ID"));

System.out.print("ISSN: " + rs.getString("ISSN\_No"));

System.out.print("Author: " + rs.getString("Author"));

System.out.print("Title: " + rs.getString("Title"));

System.out.print("Price: " + rs.getInt("Price"));

System.out.print("Publisher: " + rs.getString("Publisher"));

System.out.print("Year: " + rs.getInt("Year"));

}

}

if(choice==4)

{

Sytem.out.println("Enter the Book ID to be deleted");

int id=sc.nextInt();

String sql = "DELETE FROM Library " +

"WHERE Book\_ID = "+id+"";

stmt.executeUpdate(sql);

System.out.println("Successfully Deleted")

}

if(choice==5)

{

String sql = "SELECT \* from Library where Title LIKE \'^J\' ";

ResultSet rs = stmt.executeQuery(sql);

while(rs.next())

{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.print("ID: " + rs.getInt("Book\_ID"));

System.out.print("ISSN: " + rs.getString("ISSN\_No"));

System.out.print("Author: " + rs.getString("Author"));

System.out.print("Title: " + rs.getString("Title"));

System.out.print("Price: " + rs.getInt("Price"));

System.out.print("Publisher: " + rs.getString("Publisher"));

System.out.print("Year: " + rs.getInt("Year"));

}

}

}

}

catch(Exception e)

{

System.out.println("Failed to connect the Database");

System.out.println(e);

}

}

}