**Q 1. Write a program to implement class Employee**

import java.util.\*;

class Employee

{

int emp\_id;

String emp\_name;

float emp\_salary;

void getdata()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter id,name and salary of Employee:");

emp\_id=sc.nextInt();

emp\_name=sc.next();

emp\_salary=sc.nextFloat();

}

void putdata()

{

System.out.println("Employee Id="+emp\_id);

System.out.println("Employee Name="+emp\_name);

System.out.println("Employee Salary="+emp\_salary);

}

public static void main(String args[])

{

Employee e=new Employee();

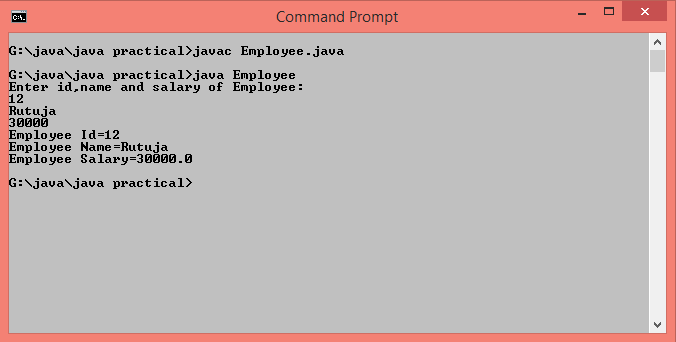
e.getdata();

e.putdata();

}

}

**Output :-**

****

**Q.2 Write a program to define date class consisting of day, month and year as data members. Write a function which compares two different Dates**

import java.util.\*;

class Date

{

int d1,d2,m1,m2,y1,y2;

void getdata()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter details of 1st date");

System.out.println("Enter day,month and year(like 31 1 2000)");

d1=sc.nextInt();

m1=sc.nextInt();

y1=sc.nextInt();

System.out.println("Enter details of 2nd date");

System.out.println("Enter day,month and year(like 31 1 2000)");

d2=sc.nextInt();

m2=sc.nextInt();

y2=sc.nextInt();

}

void compare()

{

if(d1==d2&&m1==m2&&y1==y2)

System.out.println("Dates are equal");

else

System.out.println("Dates are not equal");

}

public static void main(String args[])

{

Date d=new Date();

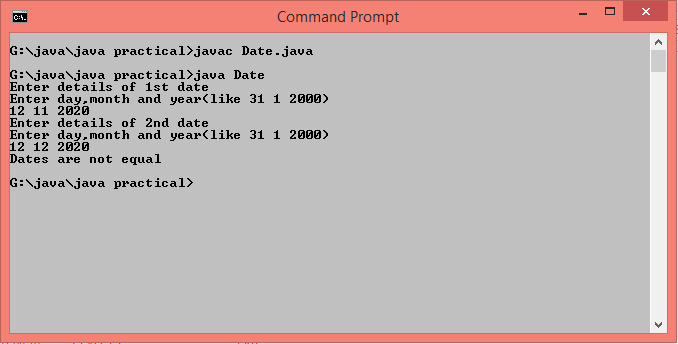
d.getdata();

d.compare();

}

}

**Output :-**

****

**Q.3 Write a program to calculate area of rectangle, circle and triangle using function overloading.**

class FunctionOverloading

{

double radius,length,breadth,base,height;

void area(int r)

{

radius=r;

System.out.println("Area of Circle="+(3.14\*radius\*radius)+" Sq.unit");

}

void area(int l,int b)

{

length=l;

breadth=b;

System.out.println("Area of Rectangle="+(length\*breadth)+" Sq.unit");

}

void area(double b,double h)

{

base=b;

height=h;

System.out.println("Area of Triangle="+(0.5\*base\*height)+" Sq.unit");

}

public static void main(String args[])

{

FunctionOverloading f=new FunctionOverloading();

f.area(5);

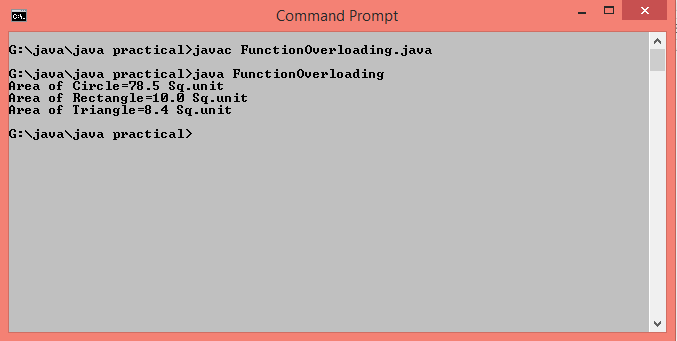
f.area(5,2);

f.area(4.2,4.0);

}

}

**Output :-**

****

**Q.4 Write a program to calculate an average of 3 integers, 4 floats and 1 array of 15 integers using function overloading**

class MethodOverloading

{

void avg(int num1,int num2,int num3)

{

System.out.println("Average of three integer values = "+((num1+num2)/3));

}

void avg(double num1,double num2,double num3,double num4)

{

System.out.println("Average of four float values = "+((num1+num2+num3+num4)/4));

}

void avg(int arr[])

{

int total=0;

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

total=total+arr[i];

}

System.out.println("Average of array is = "+(total/arr.length));

}

public static void main(String args[])

{

MethodOverloading m=new MethodOverloading();

m.avg(11,17,20);

m.avg(10.20,12.24,15.10,10.5);

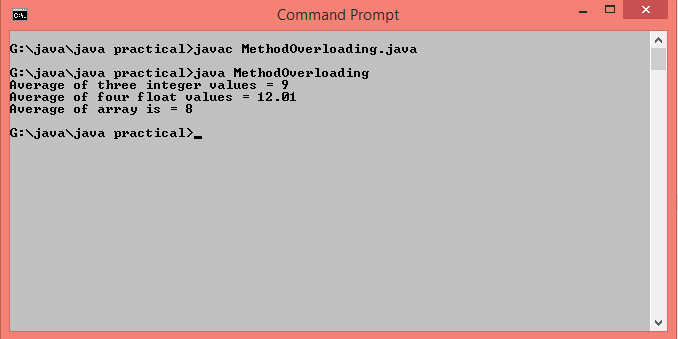
int arr[]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};

m.avg(arr);

}

}

**Output :-**

****

**Q.5 Write a program to implement "Finalize()" method.**

class FinalizeMethod

{

void show()

{

System.out.println("Example of Finalize() method");

}

protected void finalize()throws Throwable

{

try

{

System.out.println("Existing the object");

}

catch(Throwable e)

{

throw e;

}

}

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

{

FinalizeMethod f=new FinalizeMethod();

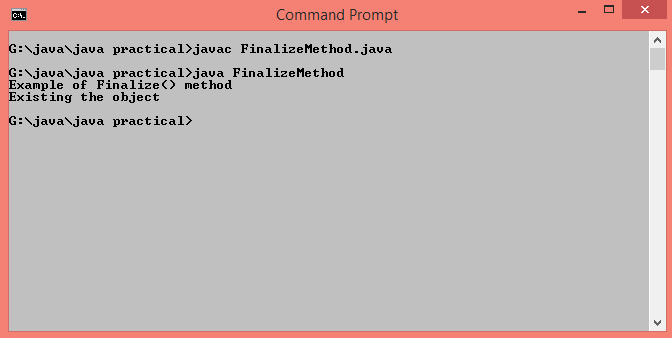
f.show();

f.finalize();

}

}

**Output :-**

****

**Q.6 Write a program to implement constructor, parameterized constructor and constructor overloading.**

class Area

{

Area()

{

System.out.println("Example of Constructor Overloading");

}

Area(int r)

{

System.out.println("Area of Circle="+(3.14\*r\*r)+" Sq.unit");

}

Area(int l,int b) {

System.out.println("Area of Rectangle="+(l\*b)+" Sq.unit");

}

}

class ConstructorOverloading

{

public static void main(String args[])

{

Area c=new Area();

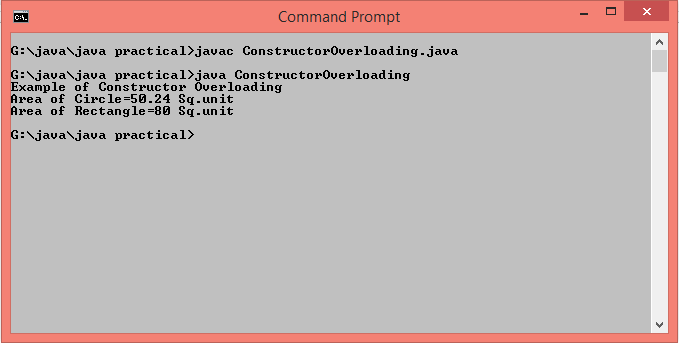
Area c1=new Area(4);

Area c2=new Area(10,8);

}

}

**Output :-**

****

**Q.7 Write a program to implement Nested, Inner Class & Anonymous Classes.**

class Outer

{

public void displayOuter()

{

System.out.println("I am from Outer Class");

Nested n=new Nested();

n.displayNested();

}

class Nested

{

public void displayNested()

{

System.out.println("I am from Nested Class");

Inner n=new Inner();

n.displayInner();

}

class Inner

{

public void displayInner()

{

System.out.println("I am from Inner Class");

}

}

}

}

interface Anonymous

{

void displayAnonymous();

}

class AllNestedClassesDemo

{

public static void main(String args[])

{

Outer o=new Outer();

o.displayOuter();

Anonymous a=new Anonymous()

{

public void displayAnonymous()

{

System.out.println("I am from Anonymous Class");

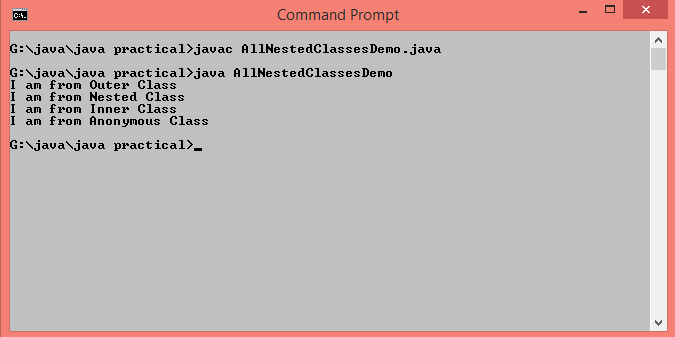
}

};

a.displayAnonymous();

}

}

**Output :-** ****

**Q.8 Write a program to implement Abstract Class.**

abstract class Vehicle

{

abstract void bike(String name);

abstract void driveBike();

}

class AbstractDemo extends Vehicle

{

void bike(String n)

{

System.out.println(" My bike name is "+n);

}

void driveBike()

{

System.out.println(" I am driving bike in pune.");

}

public static void main(String args[])

{

AbstractDemo a=new AbstractDemo();

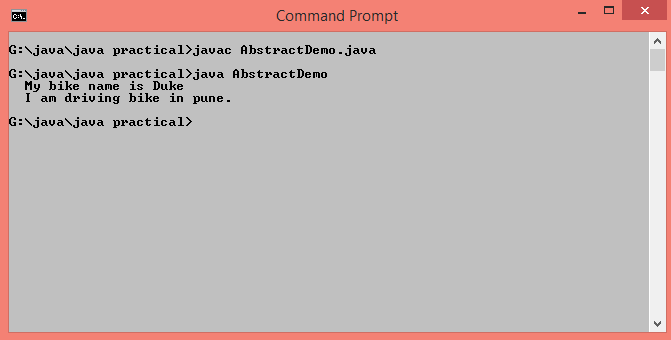
a.bike("Duke");

a.driveBike();

}

}

**Output :-**

****

**Q.9 Write a program to define Interface "CalculateArea" with method "area()". Design class to implement interface**

interface CalculateArea{

public void area();

}

class InterfaceDemo implements CalculateArea{

int radius;

InterfaceDemo(int r) {

radius=r;

}

public void area() {

System.out.println("Area of circle= "+(3.14\*radius\*radius));

}

public static void main(String args[]) {

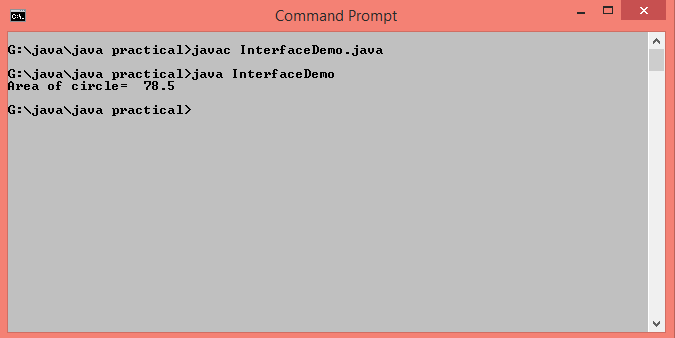
InterfaceDemo i=new InterfaceDemo(5);

i.area();

}

}

Output :-



**Q.10 Write a recursive function to calculate factorial of given number.**

class Factorial

{

static int calculate(int n)

{

if(n==0)

return 1;

else

return(n\*calculate(n-1));

}

public static void main(String args[])

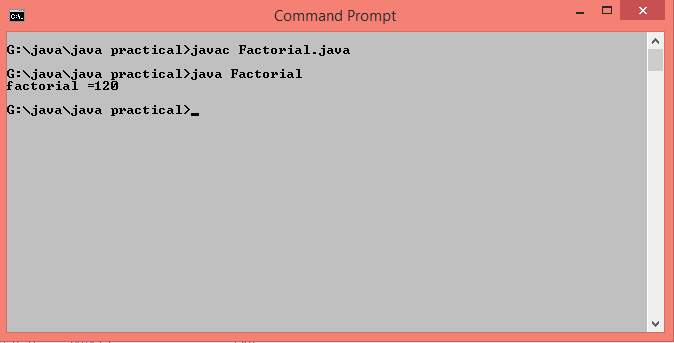
{

System.out.println("factorial ="+calculate(5));

}

}

**Output :-**

****

**Q.11 Write a program to implement "this" keyword.**

class TriangleDemo

{

int base,height;

TriangleDemo(int base,int height)

{

System.out.println("The value of Base in constructor= "+base);

System.out.println("The value of Base from global variable base= "+this.base);

this.base=base;

this.height=height;

}

public void show()

{

System.out.println(" Base= "+base+"\n Height= "+height+"\n Area of triangle = "+(0.5\*base\*height));

}

}

class ThisDemo

{

public static void main(String args[])

{

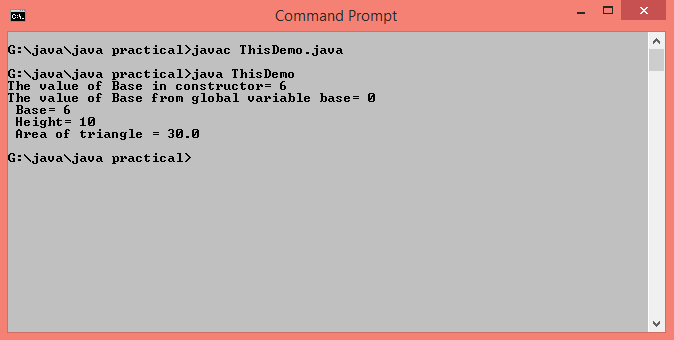
TriangleDemo t=new TriangleDemo(6,10);

t.show();

}

}

**Output :-**

****

**Q.12 Write a program to implement Accessors and Mutator methods for class "Student".**

class Student

{

private int PRN;

public int getPRN()

{

return PRN;

}

public void setPRN(int PRN)

{

this.PRN=PRN;

}

public static void main(String args[])

{

Student s=new Student();

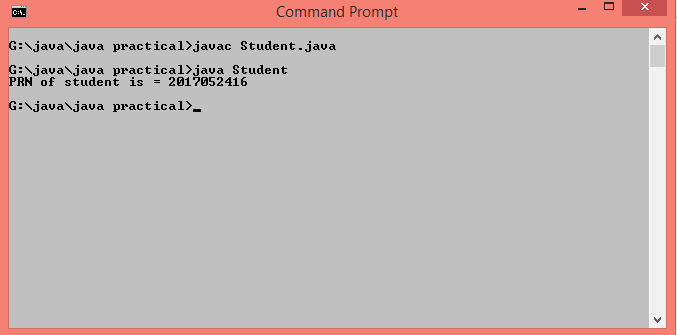
s.setPRN(2017052416);

System.out.println("PRN of student is = "+s.getPRN());

}

}

**Output :-**

****

**Q.13 Write a program to implement Generic Class.**

class People<T>

{

private T value;

public People(T value){

this.value=value;

}

public T getData(){

return this.value;

}

}

class GenericClassDemo{

public static void main(String args[]){

People<String> p1=new People<>("Santosh Kadam");

System.out.println("Name of Person : "+p1.getData());

People<Integer> p2=new People<>(40);

System.out.println("Age of Person : "+p2.getData());

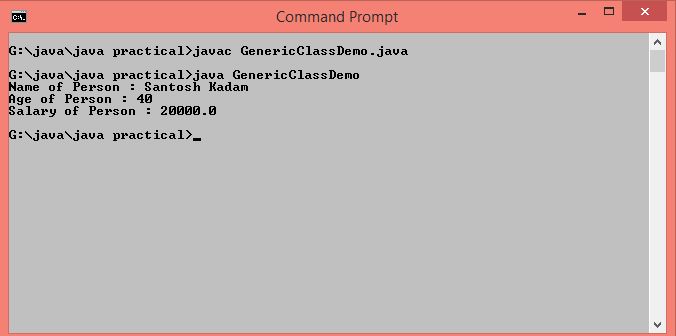
People<Double> p3=new People<>(20000.00);

System.out.println("Salary of Person : "+p3.getData());

}

}

**Output :-**

****

**Q.14 Write separate programs to implement all types of inheritance.**

**//Single Inheritance**

class Student{

int roll=5;

String name="Manthan";

int java=75;

int dsa=80;

void display()

{

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

System.out.println("Roll No : "+roll+"\nName : "+name);

}

}

class Result extends Student{

int total=java+dsa;

void show()

{

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

}

}

class SingleInheritance{

public static void main(String args[]){

Result r=new Result();

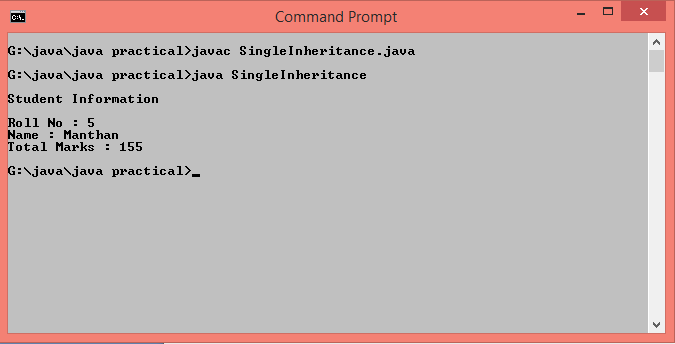
r.display();

r.show();

}

}

**Output :-**

****

**//Multilevel Inheritance**

class Student{

int roll=3;

String name="Siddhi";

int java=76;

int dsa=80;

void display()

{

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

System.out.println("Roll No : "+roll+"\nName : "+name);

}

}

class Marks extends Student{

int total=java+dsa;

void show()

{

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

}

}

class Result extends Marks{

void showResult(){

if(total>=70 && total<=200){

System.out.println("You Are Passed\nPercentage :"+(total/2));

}

else{

System.out.println("You Are Failed");

}

}

}

class MultilevelInheritance{

public static void main(String args[]){

Result r=new Result();

r.display();

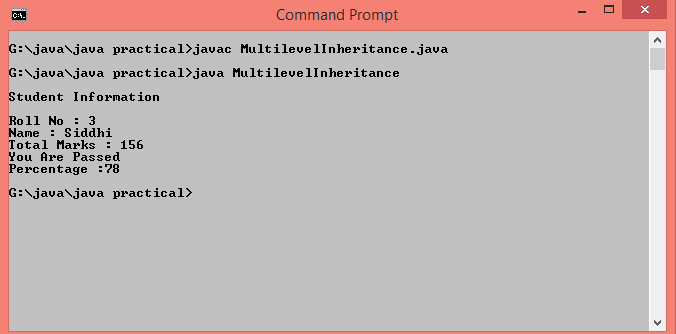
r.show();

r.showResult();

}

}

**Output :-**

****

**//Multiple Inheritance**

class Person{

int age=50;

String name="Santosh Vithoba Kadam";

void show(){

System.out.println("\n Name : "+name+"\n Age : "+age);

}

}

interface test{

public void display();

}

class Result extends Person implements test{

public void display(){

if(age>=45 && age<=100)

System.out.println(" You are eligible for Covid-19 Vaccine");

else

System.out.println(" You are not eligible for Covid-19 Vaccine");

}

}

class MultipleInheritance{

public static void main(String args[]){

Result r=new Result();

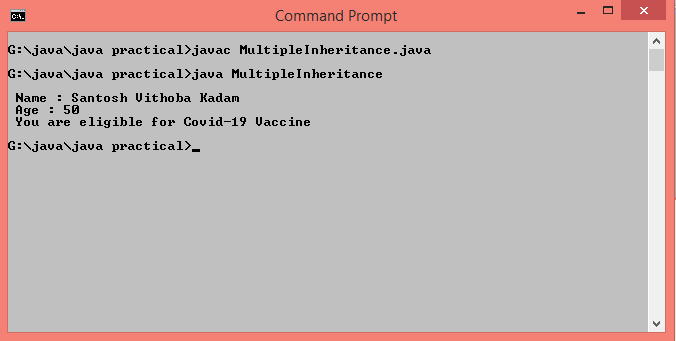
r.show();

r.display();

}

}

**Output :-**

****

**//Hierarchical Inheritance**

class Number{

int num1=153,num2=343;

void displayNumber(){

System.out.println("\n Number 1 : "+num1+"\n Number 2 : "+num2);

}

}

class Armstrong extends Number{

int temp=num1,rem,sum=0;

void showArmstrong(){

while(num1>0){

rem=num1%10;

sum=sum+(rem\*rem\*rem);

num1=num1/10;

}

if(temp==sum)

System.out.println(+temp+" is Armstrong Number.");

else

System.out.println(+temp+" is not Armstrong Number.");

}

}

class Palindrome extends Number{

int temp=num2,rem,rev=0;

void showPalindrome(){

while(num2>0){

rem=num2%10;

rev=(rev\*10)+rem;

num2=num2/10;

}

if(temp==rev)

System.out.println(+temp+" is Palindrome Number.");

else

System.out.println(+temp+" is not Palindrome Number.");

}

}

class HierarchicalInheritance{

public static void main(String[] args){

Armstrong a=new Armstrong();

a.displayNumber();

a.showArmstrong();

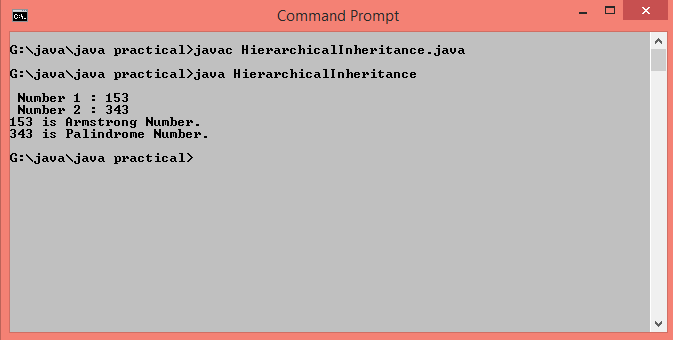
Palindrome p=new Palindrome();

p.showPalindrome();

}

}

**Output :-**

****

**//hybrid inheritance**

class Student{

int seatno=1960;

String name="Rutuja Santosh Kadam";

void display(){

System.out.println("\n Seat Number : "+seatno+"\n Name : "+name);

}

}

class JEE extends Student{

int JEEmarks=288;

void displayJEE(){

System.out.println(" JET Marks : "+JEEmarks);

}

}

interface CET{

public void displayCET();

}

class Result extends JEE implements CET{

int CETmarks=180;

public void displayCET(){

System.out.println(" CET Marks : "+CETmarks);

if(JEEmarks>=275 && JEEmarks<=300 && CETmarks>=175 && CETmarks<=200)

System.out.println("You are eligible to take admission in DY patil College.");

else

System.out.println("You are not eligible to take admission in DY patil College.");

}

}

class HybridInheritance{

public static void main(String args[]){

Result r=new Result();

r.display();

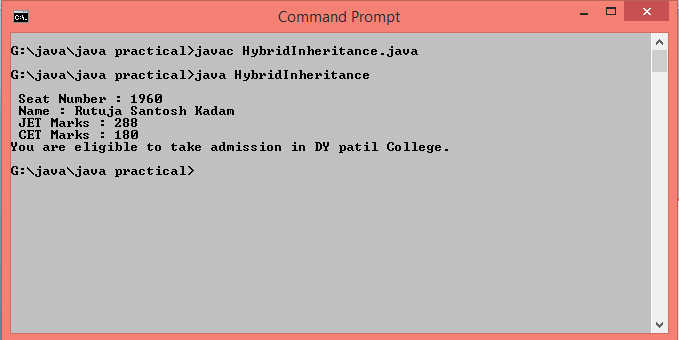
r.displayJEE();

r.displayCET();

}

}

**Output :-**

****

**Q.15 Write a program to implement multilevel inheritance with parameterized constructors defined in each class.**

class Addition{

Addition(int a,int b){

System.out.println("Addition : "+(a+b));

}

}

class Substraction extends Addition{

Substraction(int a,int b){

super(a,b);

System.out.println("Substraction : "+(a-b));

}

}

class Multiplication extends Substraction{

Multiplication(int a,int b){

super(a,b);

System.out.println("Multiplication : "+(a\*b));

}

}

class Division extends Multiplication{

Division(int a,int b){

super(a,b);

System.out.println("Division : "+(a/b));

}

}

class Multilevel{

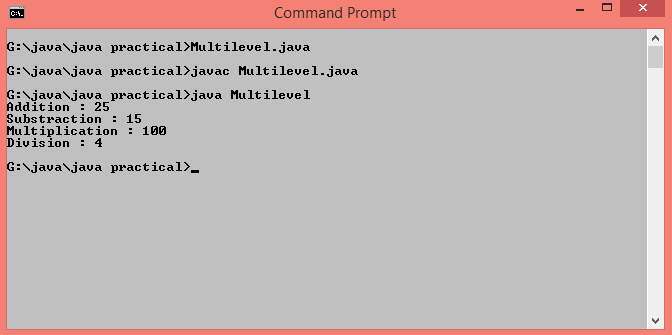
public static void main(String args[]){

Division d=new Division(20,5);

}

}

**Output :-**

****

**Q.16 Write a program to define interface "Area" for calculating area of shape. Define classes square and triangle to implement the interface "Area"**

interface Area{

public void calArea();

}

class Square implements Area{

public void calArea(){

int side=5;

System.out.println("\nArea of Square : "+(side\*side));

}

}

class Triangle implements Area{

public void calArea(){

int base=5,height=8;

System.out.println("Area of Triangle : "+(0.5\*base\*height));

}

}

class Interface{

public static void main(String args[]){

Square s=new Square();

Triangle t=new Triangle();

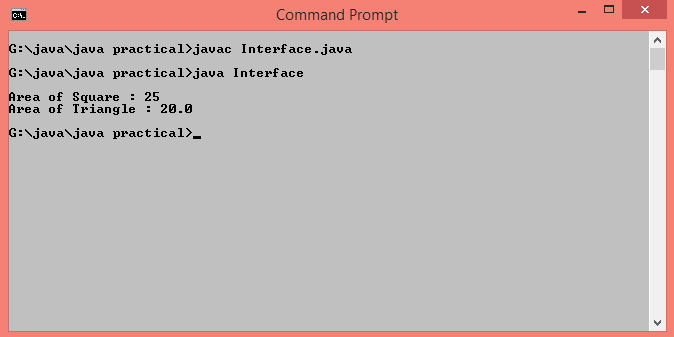
s.calArea();

t.calArea();

}

}

**Output :-**

****

**Q.17 Write class student and store it into a package. Write a class batch with information about subject, faculty and timing. Store it in package bat. Use batch class to set information in student class.**

**//Student class**

package mypack;

public class Student{

int age;

String sex;

String name;

public Student(String n,int a,String s){

name=n;

age=a;

sex=s;

}

public void display(){

System.out.println("Student Information");

System.out.println(" Name : "+name+"\n Age :"+age+"\n Sex : "+sex);

}

}

//**Batch class**

import mypack.\*;

import java.util.\*;

public class Batch{

public static void main(String args[]){

String faculty\_name="Prof.Shweta Padale",subject="Java",name,sex;

Double d=10.00;

int age;

Scanner sc=new Scanner(System.in);

System.out.println("\n Enter name,age,sex of a student :");

name=sc.next();

age=sc.nextInt();

sex=sc.next();

mypack.Student s=new mypack.Student(name,age,sex);

s.display();

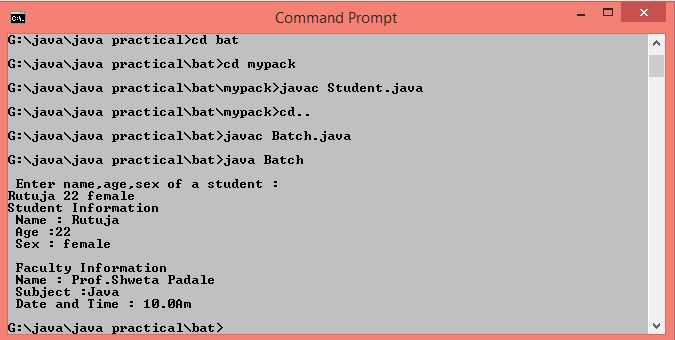
System.out.println("\n Faculty Information");

System.out.println(" Name : "+faculty\_name+"\n Subject :"+subject+"\n Date and Time : "+d+"Am");

}

}

**Output :-**

****

**Q.18 Write a program to create try block to create three types of exception and then incorporate necessary catch blocks to catch and handle them.**

class Exception{

void showException(){

try{

String s="Rutuja";

String s1=null;

int a[]=new int[10];

a[20]=100;

int n=Integer.parseInt(s);

System.out.println("Length of given String is "+s1.length());

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println("Out of size");

}

catch(NullPointerException e){

System.out.println("Cannot calculate length of null string");

}

catch(NumberFormatException e){

System.out.println("Only number required");

}

}

}

class TryCatchDemo{

public static void main(String args[]){

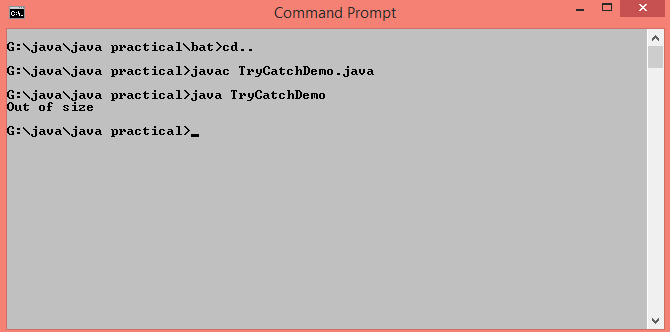
Exception e1=new Exception();

e1.showException();

}

}

**Output :-**

****

**Q.19 Write a program to accept marks for 3 subjects (Max. marks are 100 and min. Mark is 0). Implement exception handling for invalid marks.**

import java.util.\*;

class StudentMarks extends Throwable {

StudentMarks(String error)

{

super(error);

} }

public class MyException {

public static void main(String arg[]) {

try {

Scanner s=new Scanner(System.in);

System.out.print("Enter marks here : ");

int m1=s.nextInt();

int m2=s.nextInt();

int m3=s.nextInt();

if(m1<0 || m1>100 || m2<0 || m2>100 || m3<0 || m3>100) {

throw(new StudentMarks("Invalid marks"));

}

System.out.print("Entered marks are : " +m1+" "+m2+" "+m3);

}

catch(InputMismatchException e) {

System.out.println("Invalid Input..Pls Input Integer Only..");

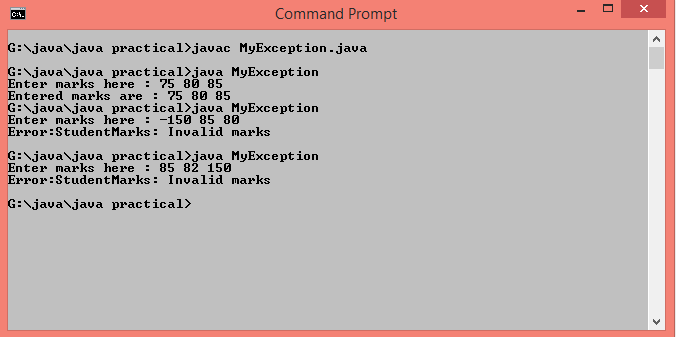
}

catch(StudentMarks e) {

System.out.println("Error:"+e);

} } }

**Output ;-**

****

**Q.20 Write a program to create thread and use getPriority and setPriority methods.**

class ThreadMin extends Thread{

public void run(){

System.out.println("Priority of thread t1(MinPriority) is: "+Thread.currentThread().getPriority());

}

}

class ThreadMax extends Thread{

public void run(){

System.out.println("Priority of thread t2 (MaxPriority) is : "+Thread.currentThread().getPriority());

}

}

class ThreadNorm extends Thread{

public void run(){

System.out.println("Priority of thread t3 (NormPriority) is : "+Thread.currentThread().getPriority());

}

}

class ThreadDefault extends Thread{

public void run(){

System.out.println("Priority of thread t4 is (Bydefault priority ) : "+Thread.currentThread().getPriority());

}

}

class JavaPriorities extends Thread{

public static void main(String args[]){

ThreadMin t1=new ThreadMin();

ThreadMax t2=new ThreadMax();

ThreadNorm t3=new ThreadNorm();

ThreadDefault t4=new ThreadDefault();

t1.setPriority(Thread.MIN\_PRIORITY);

t2.setPriority(Thread.MAX\_PRIORITY);

t3.setPriority(Thread.NORM\_PRIORITY);

t1.start();

t2.start();

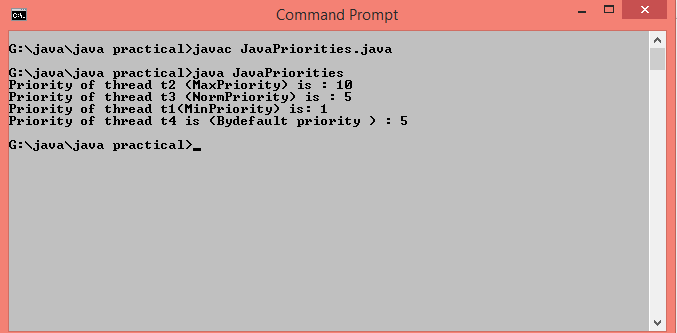
t3.start();

t4.start();

}

}

**Output :-**

****

**Q.21 Write a program to implement join method.**

class ThreadJoinMethod extends Thread{

public void run(){

for(int i=1;i<=3;i++){

try{

Thread.sleep(500);

}

catch(Throwable e){

System.out.println(e);

}

System.out.println(i);

}

}

public static void main(String args[]){

ThreadJoinMethod t1=new ThreadJoinMethod();

ThreadJoinMethod t2=new ThreadJoinMethod();

ThreadJoinMethod t3=new ThreadJoinMethod();

t1.start();

try{

t1.join();

}

catch(Throwable e){

System.out.println(e);

}

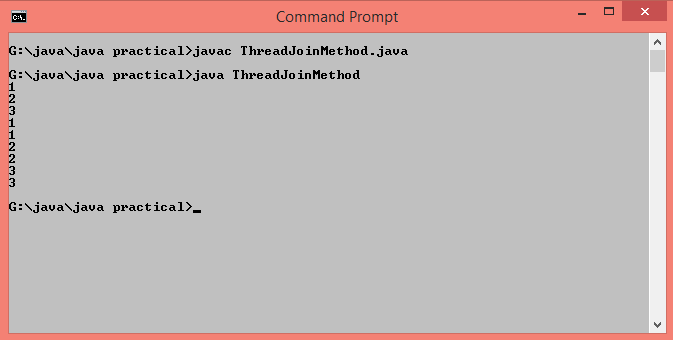
t2.start();

t3.start();

}

}

**Output :-**

****

**Q.22 Write a program to accept two integers from user using different text box and display addition, subtraction, multiplication and division in separate text boxes using applet.**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class OperationUsingApplet extends Applet implements ActionListener{

TextField num1,num2,tadd,tsub,tmult,tdiv;

Button show;

Label l1,l2,l3,l4,l5,l6;

public void init(){

l1=new Label("Enter 1st number ");

l2=new Label("Enter 2nd number ");

l3=new Label("Addition");

l4=new Label("Substraction ");

l5=new Label("Multiplication");

l6=new Label("Division");

tadd=new TextField(10);

tsub=new TextField(10);

tmult=new TextField(10);

tdiv=new TextField(10);

num1=new TextField(10);

num2=new TextField(10);

show=new Button("Show");

add(l1);

add(num1);

add(l2);

add(num2);

add(show);

add(l3);

add(tadd);

add(l4);

add(tsub);

add(l5);

add(tmult);

add(l6);

add(tdiv);

show.addActionListener(this);

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource()==show)

{

double x,y;

x=Integer.parseInt(num1.getText());

y=Integer.parseInt(num2.getText());

tadd.setText(""+(x+y));

tsub.setText(""+(x-y));

tmult.setText(""+(x\*y));

tdiv.setText(""+(x/y));

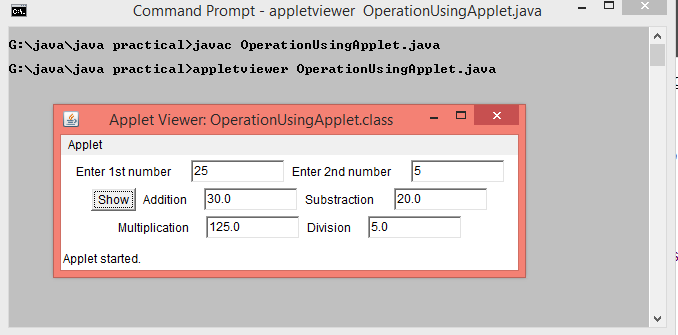
}

}

}

/\*<applet code="OperationUsingApplet.class" width="500" height="200"></applet>\*/

**Output :-**

****

**Q.23 Write a program to scroll Text from left to right.**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class ScrollingText extends Applet implements Runnable

{

String msg="Welcome to Java Programming Language .......";

Thread t=null;

public void init()

{

setForeground(Color.red);

t=new Thread(this);

t.start();

}

public void run()

{

char ch;

for(; ;)

{

try

{

repaint();

Thread.sleep(400);

ch=msg.charAt(0);

msg=msg.substring(1,msg.length());

msg+=ch;

}

catch(InterruptedException e){

System.out.println(e);

}

}

}

public void paint(Graphics g)

{

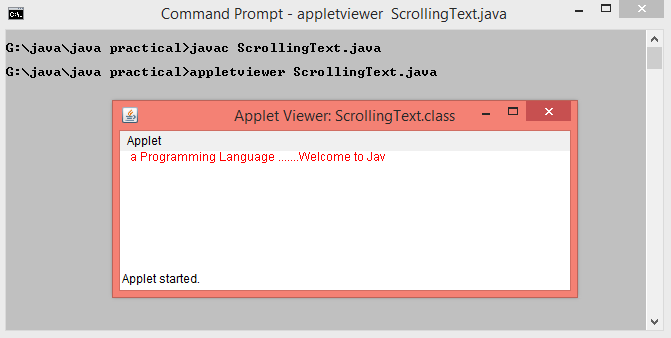
g.drawString(msg,10,10);

}

}

/\* <APPLET CODE=ScrollingText.class WIDTH=400 HEIGHT=200 > </APPLET> \*/

**Output :-**

****

**Q.24 Write an applet to display welcome message after every 5 sec. increase the font size by 2 pts.**

**Q.25 Write an applet that contains one label and one button. When the user clicks the button the associated label should reverse its text. Clicking on button should reverse the text again back to its original form.**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class ReverseStringInApplet extends Applet implements ActionListener{

Button b;

Label l;

public void init(){

l=new Label("HELLO");

b=new Button("Reverse");

add(l);

add(b);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource()==b)

{

l.setText(reverse(l.getText()));

}

}

private String reverse(String text) {

if (text.length() > 1) {

String reversed = "";

for (int i = text.length() - 1; i >= 0; i--) {

reversed += Character.toString(text.charAt(i));

}

return reversed;

}

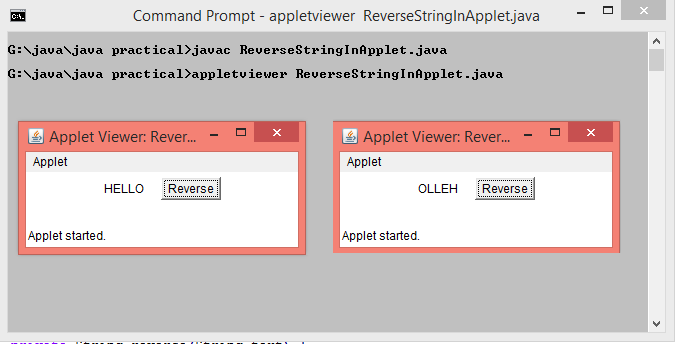
else { return text; }

}

}

/\*<applet code="ReverseStringInApplet.class" width="500" height="200"></applet>\*/

**Output :-**

****

**Q.26 Write a program to copy alternate char from one file into another.**

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class CopyFiles {

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

//Creating a File object to hold the source file

File source = new File("G:\\java\\java practical\\alternatewords1.txt");

//Creating a File object to hold the destination file

File destination = new File("G:\\java\\java practical\\alternatewords2.txt");

//Creating an FileInputStream object

FileInputStream inputStream = new FileInputStream(source);

//Creating an FileOutputStream object

FileOutputStream outputStream = new FileOutputStream(destination);

//Creating a buffer to hold the data

int length = (int) source.length();

byte[] buffer = new byte[length];

while ((length = inputStream.read(buffer)) > 0) {

outputStream.write(buffer, 0, length);

}

inputStream.close();

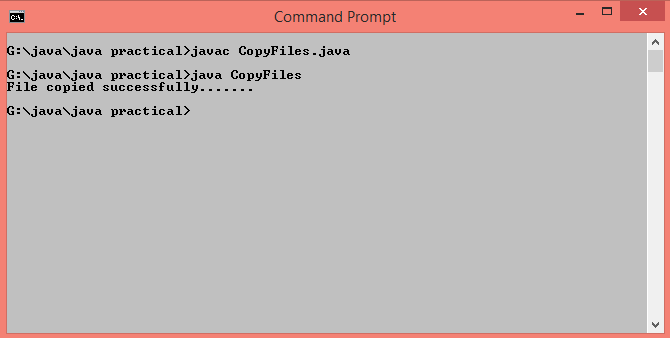
outputStream.close();

System.out.println("File copied successfully.......");

}

}

**Output :-**

****

**Q.27 Write a program to count no of words and lines into a file.**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.io.\*;

public class WordCountInFile

{

public static void main(String[] args)

{

BufferedReader reader = null;

int charCount = 0;

int wordCount = 0;

int lineCount = 0;

try

{

File file = new File("G:\\java\\java practical\\File1.txt");

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

String currentLine = reader.readLine();

while (currentLine != null)

{

lineCount++;

String[] words = currentLine.split(" ");

wordCount = wordCount + words.length;

for (String word : words)

{

charCount = charCount + word.length();

}

currentLine = reader.readLine();

}

System.out.println("Number Of Characters In a File : "+charCount);

System.out.println("Number Of Words In A File : "+wordCount);

System.out.println("Number Of Lines In A File : "+lineCount);

}

catch (IOException e)

{

e.printStackTrace();

}

finally

{

try

{

reader.close();

}

catch (IOException e)

{

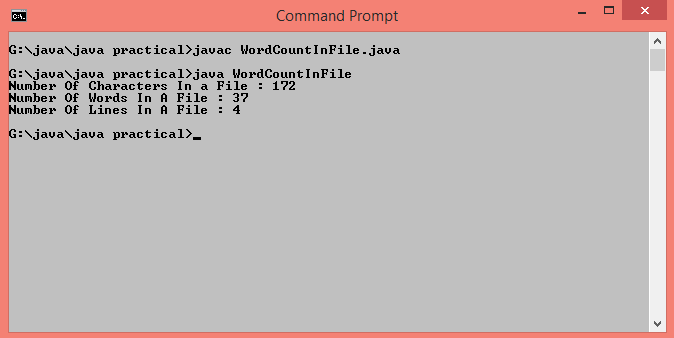
e.printStackTrace();

}

}

} }

**Output :-**

****

**Q.28 Write a program to search given word in file also display no. of occurrences in file. Accept the word to user from command line**.

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import java.util.\*;

public class SearchWordInFile

{

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

{

Scanner sc=new Scanner(System.in);

File f1=new File("alternatewords1.txt"); //Creation of File Descriptor for input file

String[] words=null; //Intialize the word Array

FileReader fr = new FileReader(f1); //Creation of File Reader object

BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object

String s;

System.out.println("Enter word to be search:");

String input;

input=sc.next();// Input word to be searched

int count=0; //Intialize the word to zero

while((s=br.readLine())!=null) //Reading Content from the file

{

words=s.split(" "); //Split the word using space

for (String word : words)

{

if (word.equals(input)) //Search for the given word

{

count++; //If Present increase the count by one

}

}

}

if(count!=0) //Check for count not equal to zero

{

System.out.println("The given word is present for "+count+ " Times in the file");

}

else

{

System.out.println("The given word is not present in the file");

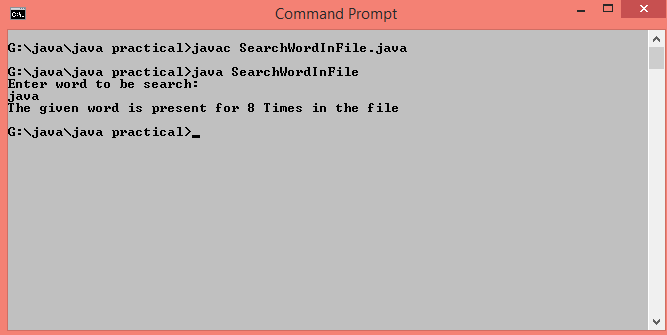
}

fr.close();

}

}

**Output :-**

****

**Q.29 Write a program to insert record for employee in to file emp.dat. The fields are id, salary.**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.StringTokenizer;

public class InsertRecord {

static void writeData(Employee e, PrintWriter out) throws IOException {

e.writeData(out);

}

static Employee readData(BufferedReader in) throws IOException {

Employee e = new Employee();

e.readData(in);

return e;

}

public static void main(String[] args) {

Employee staff = new Employee("Rutuja",5, 35500);

try {

PrintWriter out = new PrintWriter(new FileWriter("emp.dat"));

writeData(staff, out);

out.close();

} catch (IOException e) {

System.out.print("Error: " + e);

System.exit(1);

}

try {

BufferedReader in = new BufferedReader(new FileReader(

"emp.dat"));

Employee e = readData(in);

e.print();

in.close();

} catch (IOException e) {

System.out.print("Error: " + e);

System.exit(1);

}

}

}

class Employee {

private String name;

private int id;

private double salary;

public Employee(String n,int i,double s) {

name = n;

id=i;

salary = s;

}

public Employee() {

}

public void print() {

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

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

}

public void writeData(PrintWriter out) throws IOException {

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

}

public void readData(BufferedReader in) throws IOException {

String s = in.readLine();

StringTokenizer t = new StringTokenizer(s, "|");

name = t.nextToken();

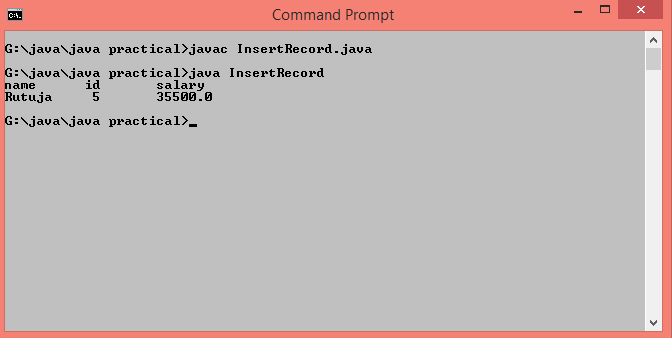
id=Integer.parseInt(t.nextToken());

salary = Double.parseDouble(t.nextToken());

}

}

**Output :-**

****

**Q.31 Write a servlet that counts number of times a client has accessed web page.**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class VisitServlet extends HttpServlet

{

static int i=1;

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws IOException, ServletException

{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String k=String.valueOf(i);

Cookie c = new Cookie("visit",k);

response.addCookie(c);

int j=Integer.parseInt(c.getValue());

if(j==1)

{

out.println("Welcome");

}

else

{

out.println("You visited "+i+" times");

}

i++;

}

}

//web file

<servlet>

<servlet-name>VisitServlet</servlet-name>

<servlet-class>VisitServlet</servlet-class>

</servlet>

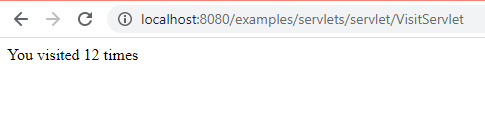
<servlet-mapping>

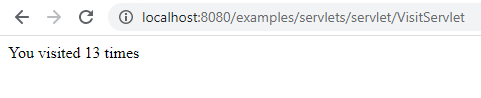
<servlet-name>VisitServlet</servlet-name>

<url-pattern>/servlets/servlet/VisitServlet</url-pattern>

</servlet-mapping>

**Output :-**

****

****