**1. Write a simple “Hello World” java program, compilation, debugging, executing using java compiler and interpreter.**

class program\_1

{

public static void main(String args[])

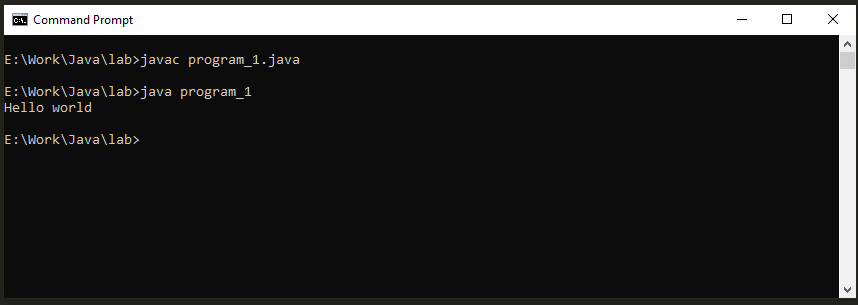
{

System.out.println("Hello world");

}

}

**Output:-**



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

public class program\_2

{

public static void main(String args[])

{

int n1=10,n2=20,i,j;

for(i=n1;i<=n2;i++)

{

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

{

if(i%j==0)

{

break;

}

}

if(j==i)

{

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

}

}

int a=0,b=n1,c;

System.out.println();

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

System.out.println(a);

System.out.println(b);

for(i=n1;i<n2-2;i++)

{

c=a+b;

System.out.println(c);

a=b;

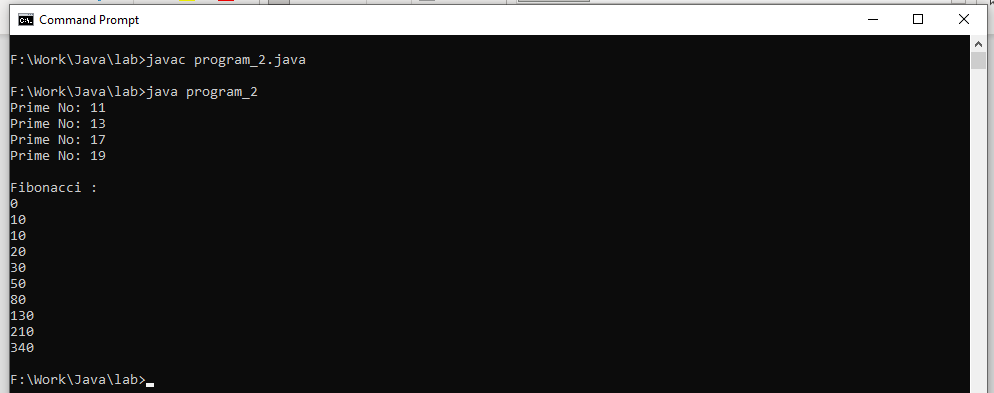
b=c;

}

}

}

**Output:-**

****

**3. Write a java program to check whether number is palindrome or not. Input: 528 Output: It is not palindrome number Input: 545 Output: It is not palindrome number.**

public class program\_3

{

public static void main(String args[])

{

int n=528,d,rev\_no=0,temp;

System.out.println(n);

temp=n;

while(n>0)

{

d=n%10;

rev\_no=rev\_no\*10+d;

n=n/10;

}

System.out.println(rev\_no);

if(rev\_no == temp)

{

System.out.println("No is Palindrom");

}

else

{

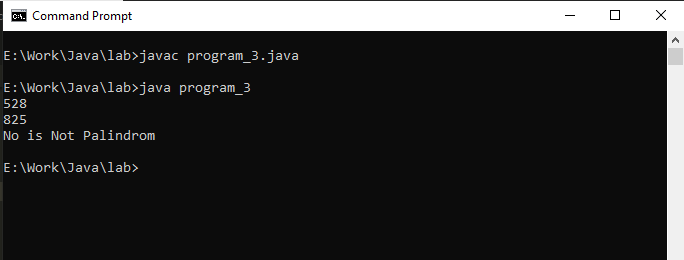
System.out.println("No is Not Palindrom");

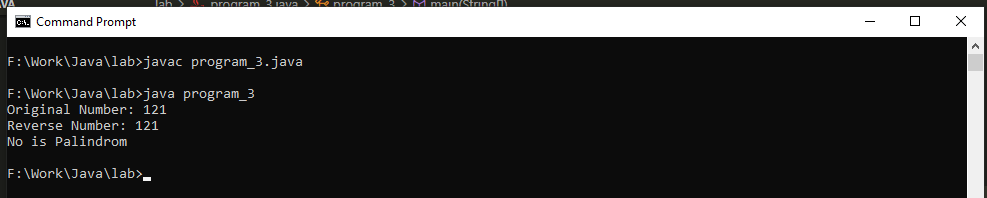
}

}

}

**Output:-**





**4. Write a java program to print value of x^n. Input: x=5 Input: n=3 Output: 125.**

public class program\_4

{

public static void main(String args[])

{

int x=5,n=3,i,ans=1;

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

{

ans=ans\*x;

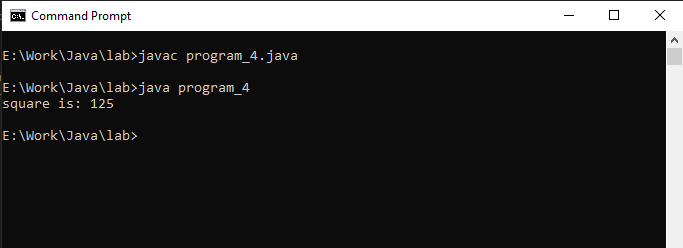
}

System.out.println("Square is: "+ans);

}

}

**Output:-**



**5. Write a java program to check Armstrong number. Input: 153 Output: Armstrong number Input: 22 Output: not Armstrong number.**

public class program\_5

{

public static void main(String args[])

{

int n=153,sum=0,d,temp;

temp=n;

while(n>0)

{

d=n%10;

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

n=n/10;

}

System.out.println(sum);

if(sum == temp)

{

System.out.println("Armstrong number");

}

else

{

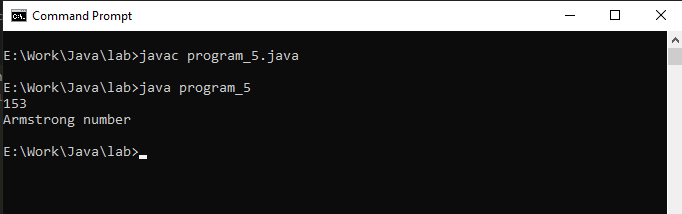
System.out.println("Not Armstrong number");

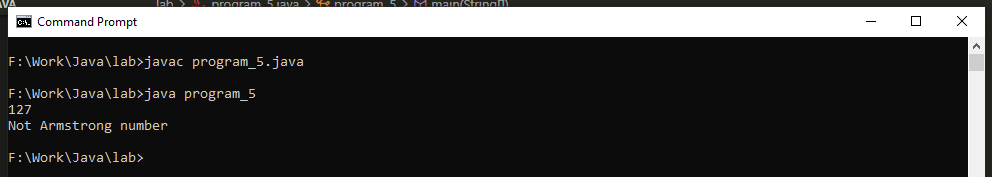
}

}

}

**Output:-**





**6. Write a program in Java to find minimum of three numbers using conditional operator.**

public class program\_6

{

public static void main(String args[])

{

int a=1,b=3,c=6;

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

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

System.out.println("C = "+c);

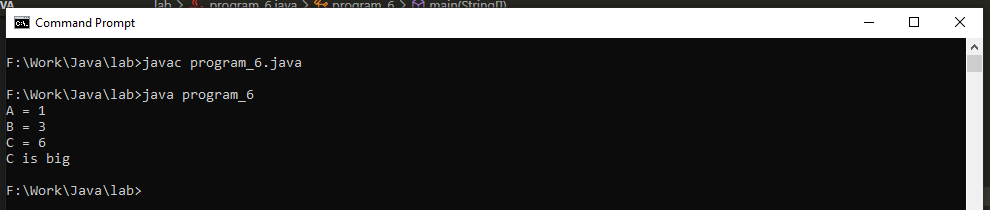
String s = c > ((a > b) ? a : b) ? "C is big" : ((a > b) ? "A is Big" : "B is big");

System.out.println(s);

}

}

**Output:-**



**7. Write a java program which should display maximum number of given 4 numbers.**

public class program\_7

{

public static void main(String args[])

{

int a=600,b=700,c=400,d=850;

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

{

System.out.println("A is big");

}

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

{

System.out.println("B is big");

}

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

{

System.out.println("C is big");

}

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

{

System.out.println("D is big");

}

else

{

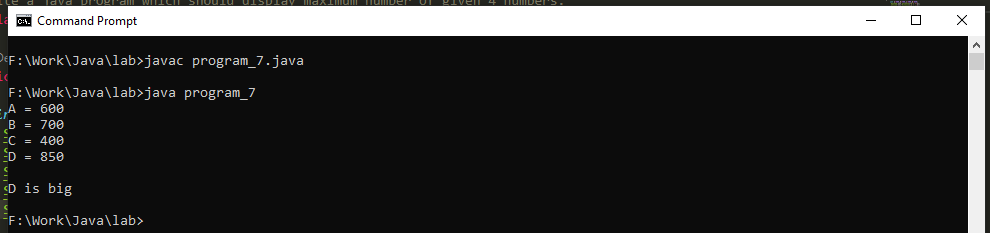
System.out.println("Put Different Value");

}

}

}

**Output:-**



**8. Write a program in Java to multiply two matrix. Declare a class Matrix where 2D array is declared as instance variable and array should be initialized, within class.**

class matrix

{

int a[][]={{1,2,4},{3,4,5},{1,2,3}};

int b[][]={{1,2,3},{4,8,6},{9,8,5}};

int c[][]=new int[3][3];

int i,j,k;

void multipy()

{

System.out.println("Matrix 1:");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

System.out.print(" "+a[i][j]);

}

System.out.println();

}

System.out.println("Matrix 2:");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

System.out.print(" "+b[i][j]);

}

System.out.println();

}

System.out.println("Multiplication of Matrix :");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

c[i][j]=0;

for(k=0;k<3;k++)

{

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

}

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

}

System.out.println();

}

}

}

class program\_8

{

public static void main(String args[])

{

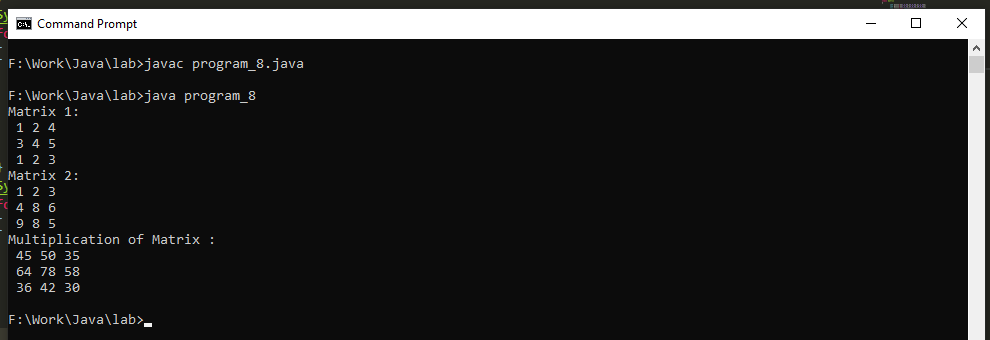
matrix ob=new matrix();

ob.multipy();

}

}

**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.**

class Matrix1

{

int a[][]={{1,2,3},{4,5,6},{7,8,9}};

int c[][]=new int[3][3];

void sum()

{

int i,j,sum;

System.out.println("Matrix 1:");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

System.out.print(" "+a[i][j]);

}

System.out.println();

}

System.out.println("Sum of Matrix Row Wise :");

for(i=0;i<3;i++)

{

sum=0;

for(j=0;j<3;j++)

{

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

}

System.out.println("Rows wise: "+sum);

}

System.out.println("Sum of Matrix Column Wise :");

for(i=0;i<3;i++)

{

sum=0;

for(j=0;j<3;j++)

{

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

}

System.out.println("Column wise: "+sum);

}

}

}

public class program\_9

{

public static void main(String args[])

{

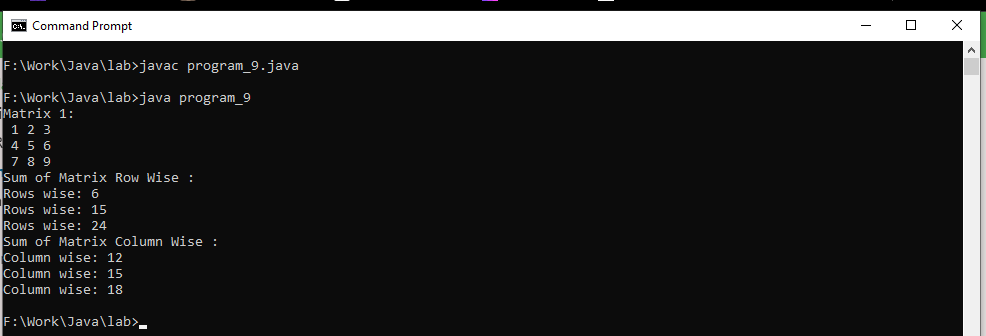
Matrix1 ob=new Matrix1();

ob.sum();

}

}

**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 = Arun**

**2.: Second Student Name is = Hiren**

**3.Third Student Name is = Hitesh**

class program\_10

{

public static void main(String args[])

{

int c=0;

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

{

c++;

System.out.println(c+". Student Name is :"+args[i]);

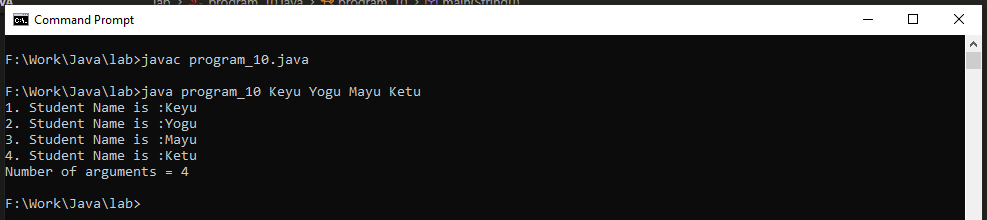
}

System.out.println("Number of arguments = "+c);

}

}

**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.**

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.IOException;

class demo

{

void countletteranddigits() throws IOException

{

int i,c=0,r=0;

InputStreamReader in=new InputStreamReader(System.in);

BufferedReader read=new BufferedReader(in);

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

String s1=read.readLine();

System.out.println("String: "+s1);

for (i = 0; i < s1.length(); i++)

{

if (Character.isLetter(s1.charAt(i)))

{

c++;

}

if(Character.isDigit(s1.charAt(i)))

{

r++;

}

}

System.out.println("Letters: "+c);

System.out.println("Digits: "+r);

}

}

class program\_11

{

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

{

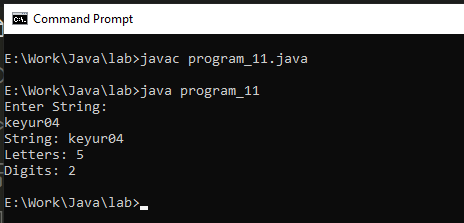
demo s1=new demo();

s1.countletteranddigits();

}

}

**Output:-**



**12.Create a class Student that would contain enrollment No, name, and gender and marks as instance variables and count as static variable which stores the count of the objects; constructors and display(). Implement constructors to initialize instance variables.**

**Output:-**

class student

{

static int eno=1,marks=50;

static String name="maya",gender="Male";

student(int e,int m,String n,String g)

{

eno=e;

marks=m;

name=n;

gender=g;

}

void display()

{

System.out.println("enrollment no: "+eno);

System.out.println("Name: "+name);

System.out.println("Gender: "+gender);

System.out.println("Marks: "+marks);

}

}

public class program\_12

{

public static void main(String args[])

{

int e=13,m=95;

String n="Keyur",g="Male";

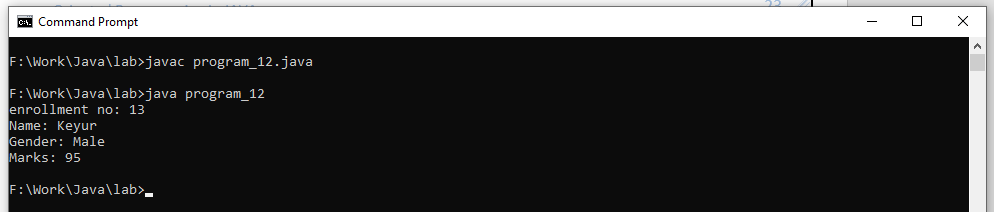
student s=new student(e, m, n, g);

s.display();

}

}

**Output:-**



**13.Write a program in Java to demonstrate use of this keyword. Check whether this can access the Static variables of the class or not.**

class student1

{

static int eno,marks;

static String name,gender="Female";

student1(int eno,int marks,String name,String g)

{

this.eno=eno;

this.marks=marks;

this.name=name;

this.gender=g;

this.display();

}

void display()

{

System.out.println("enrollment no: "+eno);

System.out.println("Name: "+name);

System.out.println("Gender: "+gender);

System.out.println("Marks: "+marks);

}

}

public class program\_13

{

public static void main(String args[])

{

int e=13,m=60;

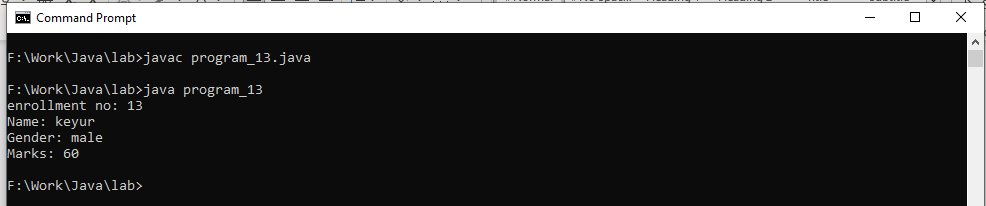
String n="keyur",g="male";

student1 s=new student1(e,m,n,g);

}

}

**Output:-**



**14.Create a class “Rectangle” that would contain length and width as an instance variable and count as a static variable. Define constructors [constructor overloading (default, parameterized and copy)] to initialize variables of objects. Define methods to find area and to display variables’ value of objects which are created. [Note: define initializer block, static initializer block and the static variable and method. Also demonstrate the sequence of execution of initializer block and static initialize block].**

class Rectangle

{

static int l,b;

static int rect;

Rectangle()

{

l=1;

b=3;

}

Rectangle(int l,int b)

{

this.l=l;

this.b=b;

}

Rectangle(Rectangle obj)

{

obj.l=10;

obj.b=20;

}

void display()

{

rect=l\*b;

System.out.println("Area of Rectangle: "+rect);

}

}

public class program\_14

{

public static void main(String args[])

{

Rectangle r = new Rectangle();

r.display();

Rectangle r1=new Rectangle(10,20);

r1.display();

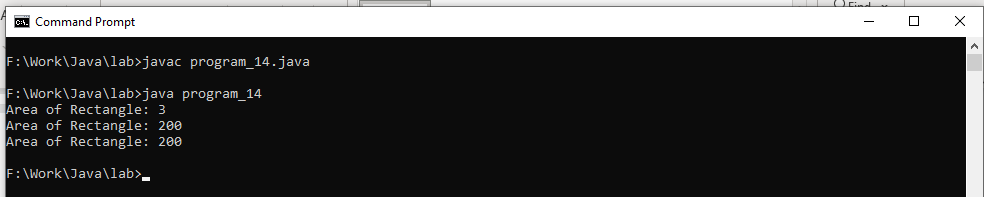
Rectangle r2=new Rectangle(r);

r2.display();

}

}

**Output:-**



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

class staticblock

{

static

{

System.out.println("I am static block 1");

}

static

{

System.out.println("I am static block 2");

}

staticblock()

{

System.out.println("I am Constractor");

}

}

public class program\_15 {

public static void main(String args[])

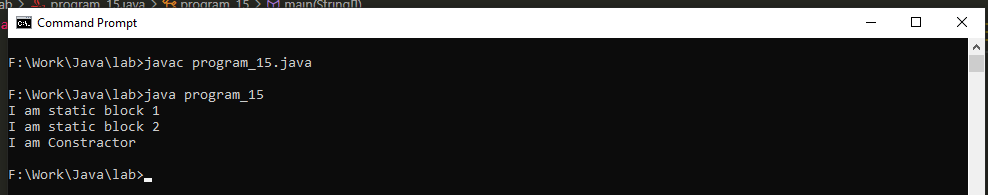
{

staticblock ob=new staticblock();

}

}

**Output:-**



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

package program\_16;

public class program\_16 {

public static void main(String args[])

{

byte a = 1;

Byte A=a;

int b = 10;

Integer B=b;

float c = 18.6f;

Float C=c;

double d = 250.5;

Double D=d;

char e='a';

Character E=e;

System.out.println("");

System.out.println("Wrapper class of each primitive data types");

System.out.println("");

System.out.println("byte to Byte value : " + A);

System.out.println("int to Integer value : " + B);

System.out.println("float to Float value : " + C);

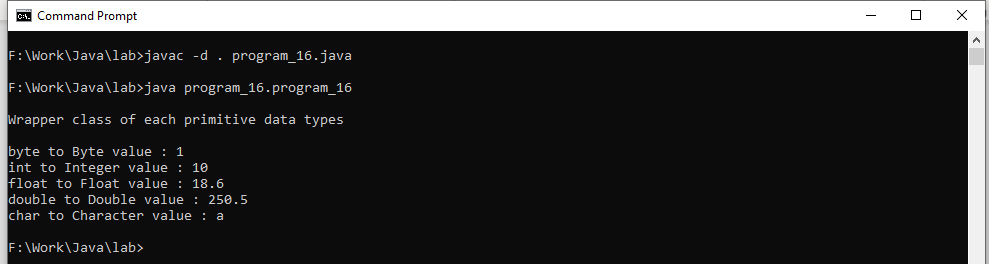
System.out.println("double to Double value : " + D);

System.out.println("char to Character value : " + E);

}

}

**Output:-**



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

class circle

{

int a=10;

int b=20;

circle()

{

System.out.println("circle radius is "+b);

}

}

class A extends circle

{

public A()

{

System.out.println("A values is "+a);

}

A(int a)

{

this();

System.out.println("value of A constractor is "+a);

}

}

class program\_17

{

public static void main(String args[])

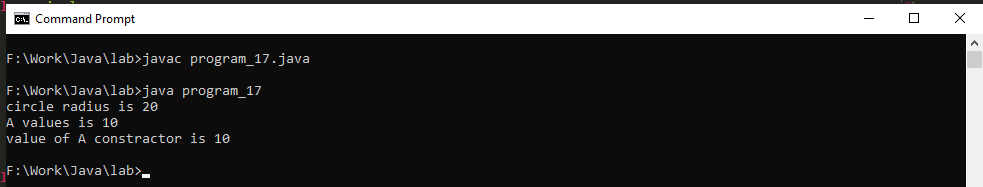
{

A a=new A(10);

}

}

**Output:-**



**18. Create a class “Vehicle” with instance variable vehicle\_type. Inherit the class in a class called “Car” with instance model\_type, company name etc. display the information of the vehicle by defining the display() in both super and sub class [ Method Overriding].**

class Vehicle

{

String vehicle\_type="two wheel";

void display()

{

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

}

}

class Car extends Vehicle

{

String model\_type="sport";

String company\_name="palser";

public void display()

{

System.out.println("Model type is:"+model\_type);

System.out.println("Company type is:"+company\_name);

}

}

class program\_18

{

public static void main(String args[])

{

Vehicle v=new Vehicle();

v.display();

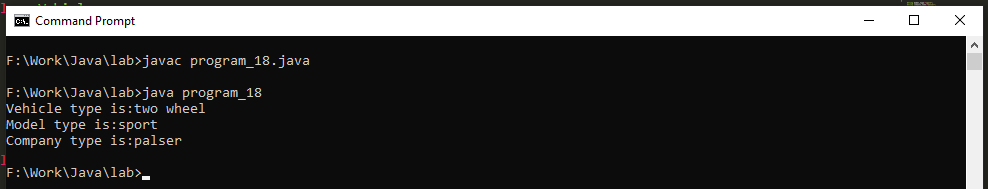
Car i=new Car();

i.display();

}

}

**Output:-**



**19. Create a class “Account” containing accountNo, and balance as an instance variable. Derive the Account class into two classes named “Savings” and “Current”. The “Savings” class should contain instance variable named interestRate, and the “Current” class should contain instance variable called overdraftLimit. Define appropriate methods 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.]**

package program\_19;

import java.io.InputStreamReader;

import java.io.BufferedReader;

import java.io.IOException;

class account {

int ano, bal;

}

class Savings extends account {

int interestRate;

void check(int depo, int wid) throws IOException {

ano = 13;

bal = 5000;

System.out.println("accountNo is " + ano);

System.out.println("balance is " + bal);

if (bal == 0) {

bal = bal + depo;

} else if (bal > 0) {

System.out.println("1. deposit");

System.out.println("2. withdraw");

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

InputStreamReader in = new InputStreamReader(System.in);

BufferedReader read = new BufferedReader(in);

int a = Integer.parseInt(read.readLine());

if (a == 1) {

System.out.println("deposit Amount is " + depo);

bal = bal + depo;

System.out.println("Current Balance is " + bal);

} else if (a == 2) {

System.out.println("withdraw value is " + wid);

int withdraw = bal - wid;

System.out.println("Balance after withdraw value " + withdraw);

} else {

System.out.println("Enter proper Choices: ");

}

}

}

}

class Current extends account {

int overdraftLimit;

void check(int depo, int overdraftLimit) throws IOException {

this.overdraftLimit = overdraftLimit;

ano = 13;

bal = 5000;

System.out.println("accountNo is " + ano);

System.out.println("balance is " + bal);

if (bal == 0) {

bal = bal + depo;

} else if (bal > 0) {

System.out.println("1. deposit");

System.out.println("2. withdraw");

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

InputStreamReader in = new InputStreamReader(System.in);

BufferedReader read = new BufferedReader(in);

int a = Integer.parseInt(read.readLine());

if (a == 1) {

System.out.println("deposit Amount is " + depo);

bal = bal + depo;

System.out.println("Current Balance is " + bal);

} else if (a == 2) {

if (overdraftLimit <= 500) {

System.out.println("withdraw value is " + overdraftLimit);

int withdraw = bal - overdraftLimit;

System.out.println("Balance after withdraw value " + withdraw);

} else {

System.out.println("over overdraftLimit Is below 500rs");

}

} else {

System.out.println("Enter proper Choices: ");

}

}

}

}

class program\_19 {

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

InputStreamReader in = new InputStreamReader(System.in);

BufferedReader read = new BufferedReader(in);

System.out.println("1. Savings Account");

System.out.println("2. Current Account");

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

int o = Integer.parseInt(read.readLine());

if (o == 1) {

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

System.out.println("Savings Account");

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

Savings a = new Savings();

a.check(2000, 100);

} else if (o == 2) {

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

System.out.println("Current Account");

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

Current b = new Current();

b.check(2000, 400);

} else {

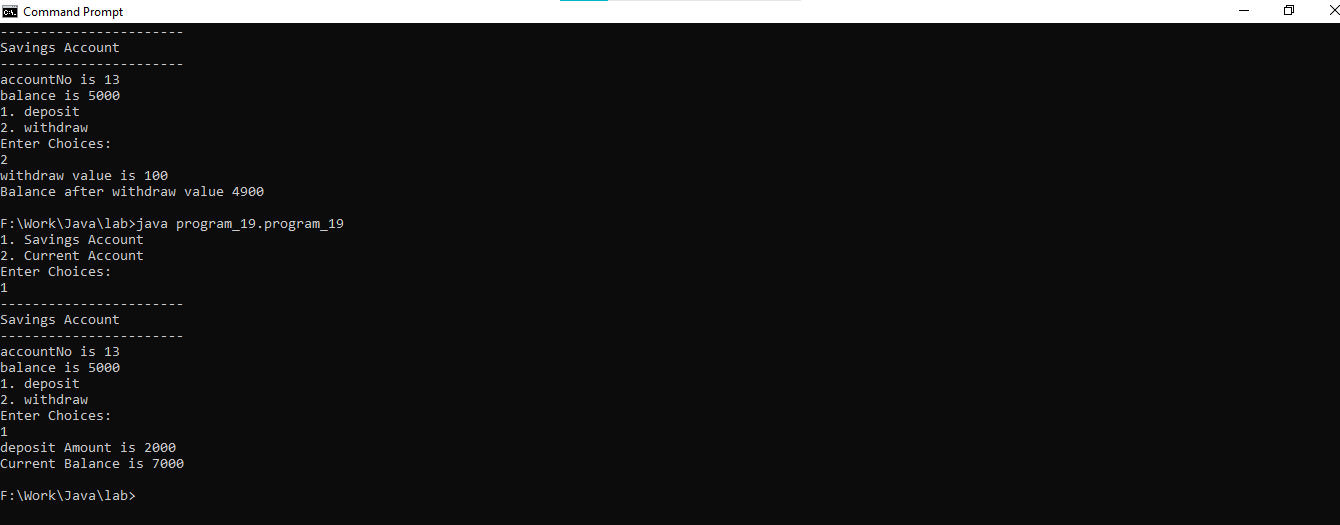
System.out.println("Enter proper Choices: ");

}

}

}

**Output:-**

****

**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].**

package program\_20;

class Account

{

Account()

{

System.out.println("Main Account");

}

}

class savingAccount extends Account{

savingAccount()

{

// by default compailer write super() to call the perent class constractor.

System.out.println("savingAccount Constuctor");

}

}

class CurrentAccount extends Account

{

CurrentAccount()

{

// by default compailer write super() to call the perent class constractor.

System.out.println("CurrentAccount Constuctor");

}

}

public class program\_20 {

public static void main(String[] args) {

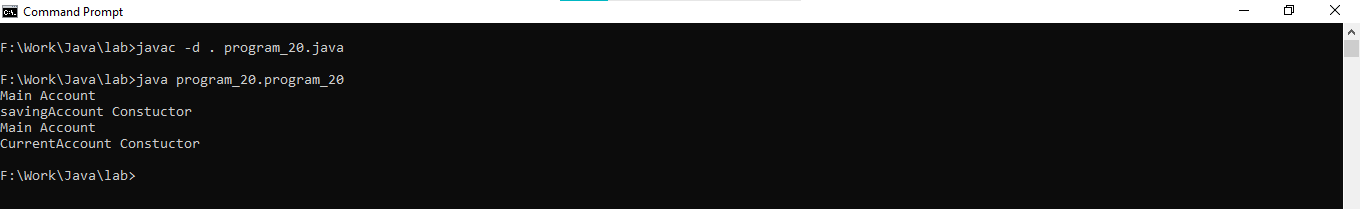
savingAccount s=new savingAccount();

CurrentAccount c=new CurrentAccount();

}

}

**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.**

class myclass{

final short k=4;//can't reassing or change the final value

byte f=20;

// void display()

// {

// k=20;

// }

}

public class program\_21 {

public static void main(String args[])

{

myclass ob=new myclass();

System.out.println("final Variable : "+ob.k);

System.out.println("Non final method: "+ob.f);

}

}

**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.**

class A{

static int x;

}

class B extends A

{

int x;

void display()

{

System.out.println("A class X variable "+super.x);

System.out.println("B class X variable "+x);

}

}

class program\_22

{

public static void main(String args[])

{

//static variable Assign using class name

A.x=10;

//non-static variable assign using class object

B b=new B();

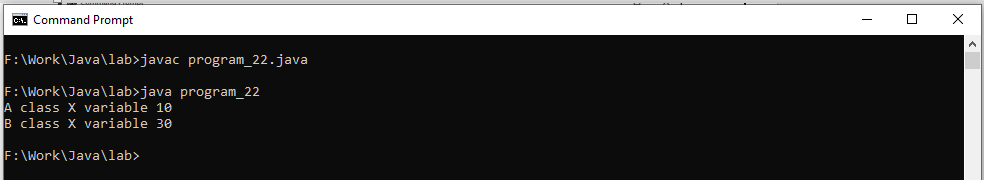
b.x=30;

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.**

abstract class Shape

{

void area()

{

};

}

class Triangle extends Shape

{

double b=30,h=20;

void area()

{

System.out.println("area() of triangle: "+0.5\*b\*h);

}

}

class Rectangle extends Shape

{

double w=20,b=30;

void area()

{

System.out.println("area() of Rectangle: "+w\*b);

}

}

class Circle extends Shape

{

double r=50;

void area()

{

System.out.println("area() of Circle: "+3.14\*(r\*r));

}

}

public class program\_23 {

public static void main(String args[])

{

Triangle t=new Triangle();

t.area();

Rectangle r=new Rectangle();

r.area();

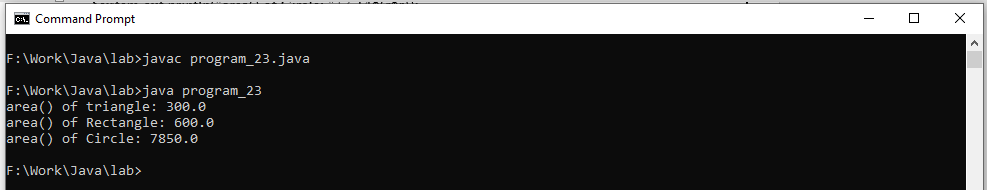
Circle c=new Circle();

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.**

import java.io.InputStreamReader;

import java.io.BufferedReader;

import java.io.IOException;

interface exam

{

public boolean pass(int mark);

}

interface classify

{

public 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>=50 && average<60)

{

return "Second";

}

else if(average<50)

{

return "No division";

}

else

{

return "Enter Proper average";

}

}

}

class program\_24

{

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

{

result ob=new result();

InputStreamReader in=new InputStreamReader(System.in);

BufferedReader read=new BufferedReader(in);

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

int mark=Integer.parseInt(read.readLine());

ob.pass(mark);

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

int average=Integer.parseInt(read.readLine());

ob.division(average);

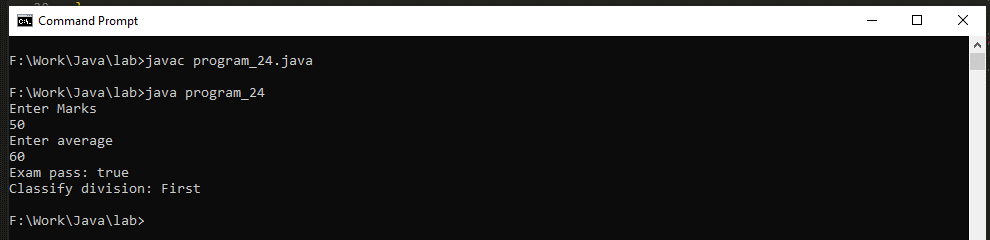
System.out.println("Exam pass: "+ob.pass(mark));

System.out.println("Classify division: "+ob.division(average));

}

}

**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.**

package student;

public class student {

int rno, results;

String name;

public student(int rno, int results, String name) {

this.rno = rno;

this.name = name;

this.results = results;

}

public void display() {

System.out.println("My Roll No:" + rno);

System.out.println("My Name is: " + name);

System.out.println("My Result is: " + results);

}

}

package exam;

import studentinfo.student;

class result {

public static void main(String args[]) {

int rno = 13;

int result = 60;

String nm = "Keyur";

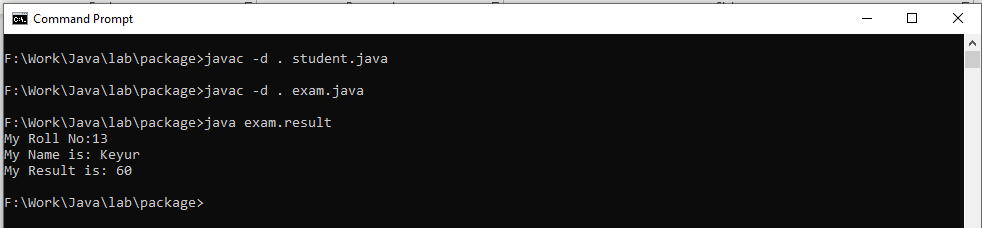
student s = new student(rno, result, nm);

s.display();

}

}

**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.**

package apack;

public class A

{

public String pub;

protected String pro;

private String pri;

}

package bpack;

import apack.A;

public class B extends A

{

public void display()

{

B ob=new B();

ob.pub="keyur";

// ob.pri="yogi";

System.out.println("public Variable "+ob.pub);

// System.out.println("public Variable "+ob.pri);

}

}

package cpack;

import apack.A;

public class C extends A{

public void display()

{

C ob=new C();

ob.pro="mayuri";

System.out.println("protected in c "+ob.pro);

}

}

package dpack;

import bpack.B;

import cpack.C;

public class ProtectedDemo

{

public static void main(String[] args) {

B ob=new B();

ob.display();

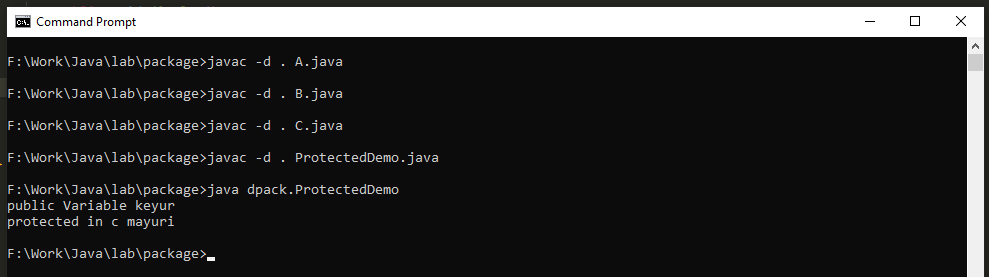
C ob1=new C();

ob1.display();

}

}

**Output:-**

****

**27. Write a java program to implement Generic class Number\_1 for both data type int and float in java.**

class Number\_1<T>

{

T data;

Number\_1(T data)

{

this.data=data;

}

T display()

{

return this.data;

}

}

class program\_27

{

public static void main(String args[])

{

int n=13;

Number\_1<Integer> ob=new Number\_1<Integer>(n);

System.out.println(ob.display());

Float n1=(float)n;

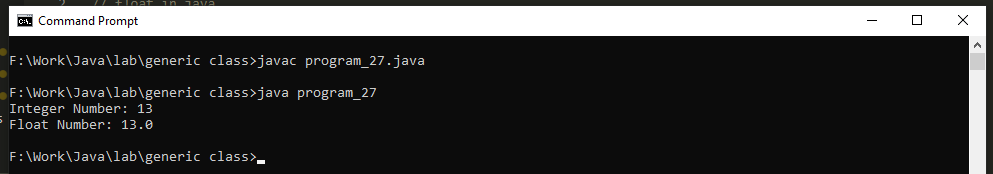
Number\_1<Float> ob1=new Number\_1<Float>(n1);

System.out.println(ob1.display());

}

}

**Output:-**



**28. Write a java program to accept string to check whether it is in Upper or Lower case. After checking, case will be reversed.**

import java.io.InputStreamReader;

import java.io.BufferedReader;

import java.io.IOException;

public class program\_28 {

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

InputStreamReader in = new InputStreamReader(System.in);

BufferedReader read = new BufferedReader(in);

System.out.print("Enter String: ");

String s1 = read.readLine();

char a[] = s1.toCharArray();

int u = 0;

int l = 0;

String rev="";

for (char k : a) {

if (k >= 'A' && k <= 'Z') {

u++;

} else if (k >= 'a' && k <= 'z') {

l++;

}else{

continue;

}

}

// System.out.println("String is: "+s1);

System.out.println("");

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

System.out.println("Uppercase letter is: " + u);

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

System.out.println("Lowercase letter is: " + l);

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

System.out.println("");

// String rev = new StringBuffer(s1).reverse().toString();

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

{

rev=rev+a[i];

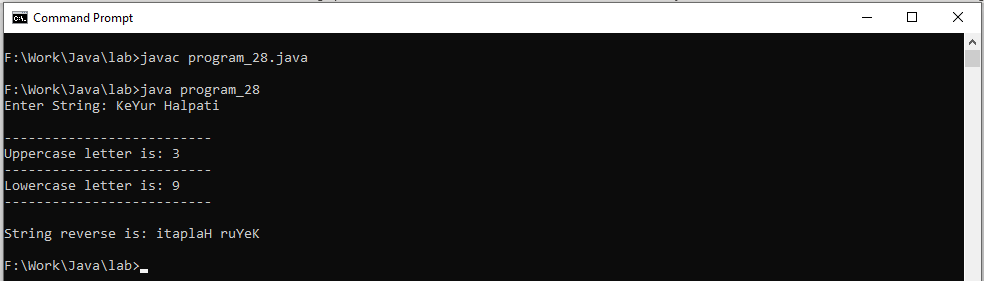
}

System.out.println("String reverse is: " + rev);

}

}

**Output:-**



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

package program\_29;

public class program\_29 {

public static void main(String args[])

{

String s1=" Halpati Keyur M";

System.out.println("");

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

System.out.println("");

System.out.println("This is String on which all operation Are perform : "+s1);

System.out.println("");

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

System.out.println("");

System.out.println("String Uppercase : "+s1.toUpperCase());

System.out.println("String LowerCase : "+s1.toLowerCase());

System.out.println("String Trim : "+s1.trim());

System.out.println("String Length : "+s1.length());

System.out.println("String Fetch Character from String: "+s1.charAt(9));

System.out.println("String Replace : "+s1.replace("Keyur", "Keyu"));

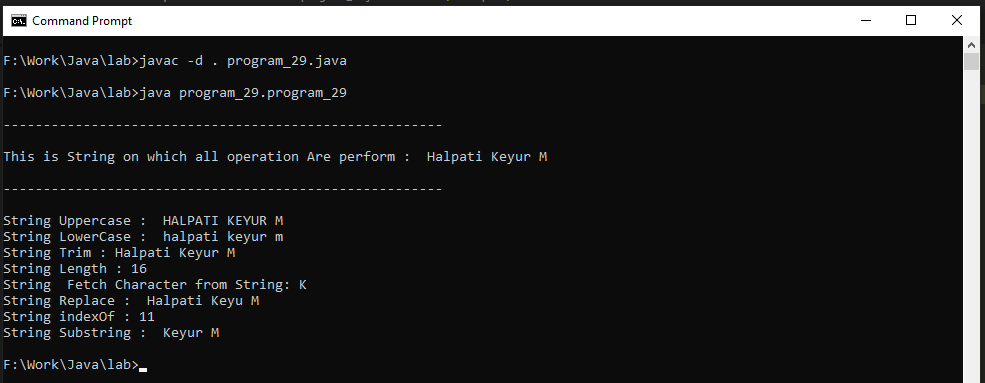
System.out.println("String indexOf : "+s1.indexOf("y"));

System.out.println("String Substring : "+s1.substring(8));

}

}

**Output:-**



**30. Write a program in Java to demonstrate use of final class, final variable and final method.**

package program\_30;

final class abc

{

final int a=10;

final void display()

{

System.out.println("Final variable: "+a);

}

}

class program\_30

{

public static void main(String[] args) {

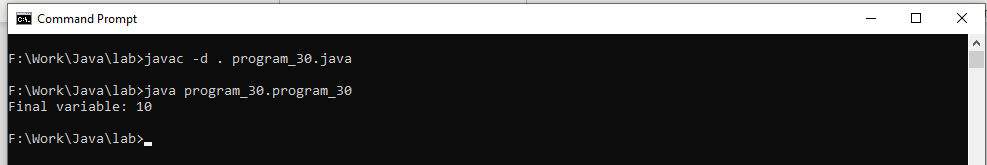
abc o=new abc();

o.display();

}

}

**Output:-**



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

class program\_31 {

public static void main(String args[]) {

int a = 10;

int b = 0;

try {

System.out.println("before array assign value");

int c = a / b;

System.out.println("after operation: ");

} catch (Throwable p) {

System.out.println("Divide by zero");

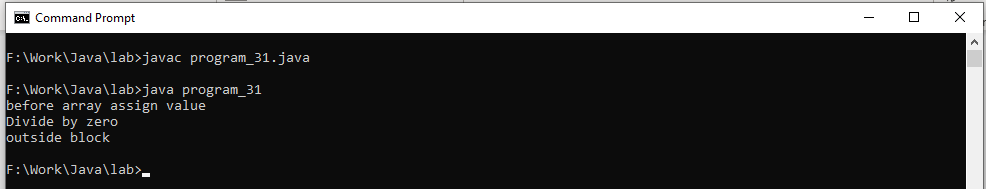
}

System.out.println("outside block");

}

}

**Output:-**



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

public class program\_32 {

public static void main(String args[]) {

int a[] = { 4, 2, 27, 2, 5, 6 };

int b[] = { 2, 0, 3, 2 };

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

try {

try {

System.out.println(" a[" + a[i] + "] /" + " b[" + b[i] + "]" + " = " + a[i] / b[i]);

// throw new Exception();

if(a[i]==b[i])

{

throw new Exception();

}

} catch (ArithmeticException e) {

System.out.println(" Divide by zero");

}

} catch (ArrayIndexOutOfBoundsException r) {

System.out.println(" Array out of Bounds");

}

catch(Exception o)

{

System.out.println("Throw Same Data");

}

finally

{

System.out.println("Exit try blocks");

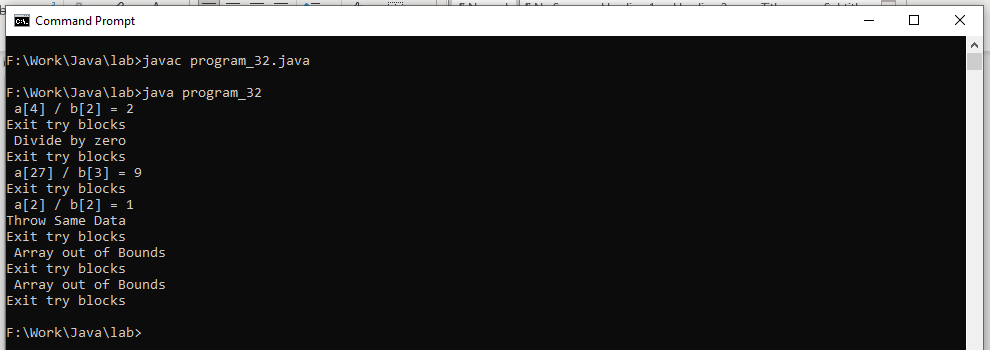
}

}

}

}

**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.**

import java.io.InputStreamReader;

import java.io.BufferedReader;

import java.io.IOException;

public class program\_33 {

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

{

InputStreamReader in=new InputStreamReader(System.in);

BufferedReader read=new BufferedReader(in);

// System.out.println("Enter deposits the amount ");

// double bal=Double.parseDouble(read.readLine());

double bal=1000.00;

System.out.println("");

System.out.println("The amount Balance is Rs "+bal);

System.out.println("Enter withdrawing Value");

double wid=Double.parseDouble(read.readLine());

try{

bal=bal-wid;

if(bal<=500.0)

{

throw new Throwable();

}

System.out.println("");

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

System.out.println("Balance is "+bal);

}catch(Throwable e)

{

System.out.println("");

System.out.println("Not Sufficient Fund");

// break;

}

finally

{

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

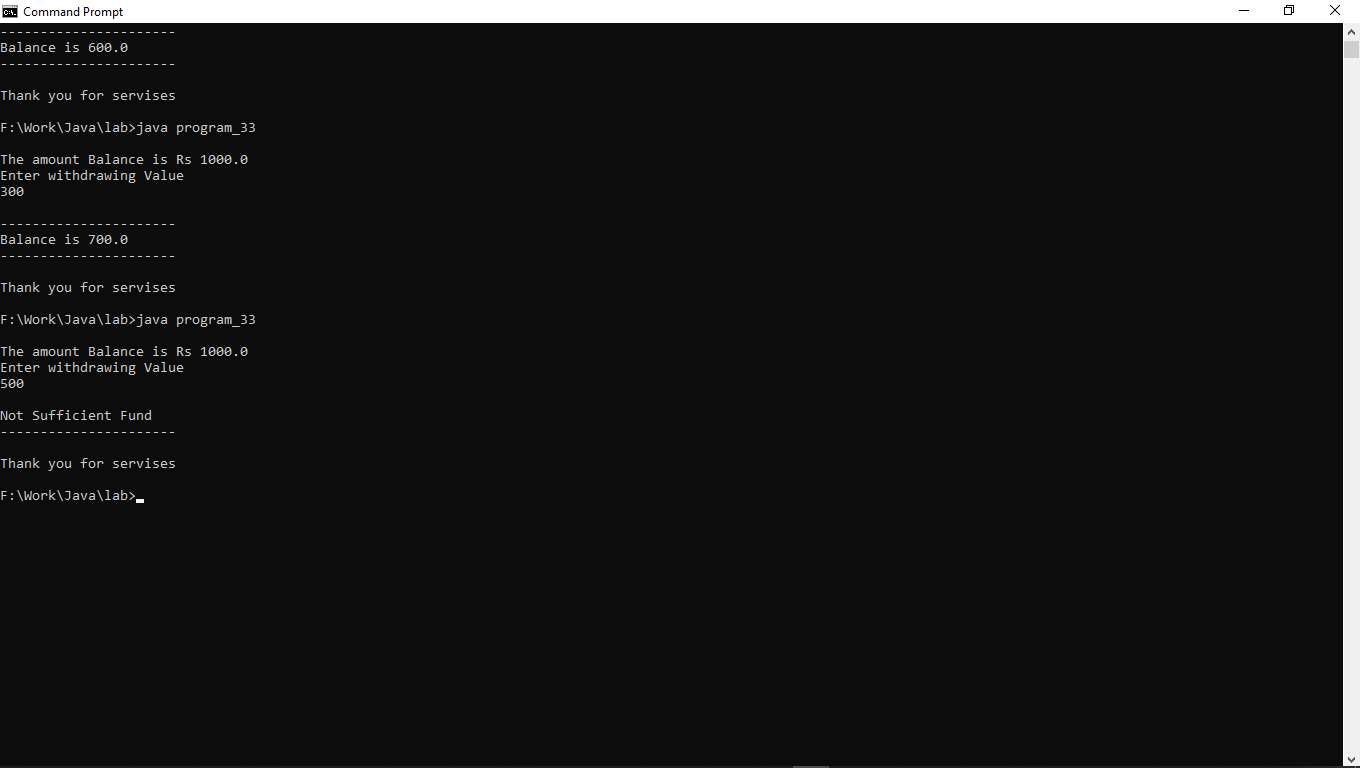
System.out.println("");

System.out.println("Thank you for servises");

}

}

}

**Output:-**

**34.Write a program to write at least 10 objects of the Circle class in a File and to perform basic operations: adding, retrieving, updating, removing elements.**

package program\_34;

import java.io.\*;

import java.util.\*;

class Circle implements Serializable

{

double id=0;

double radius=0,area=0;

Circle()

{

this.id=0;

this.radius=0;

}

Circle(double i,double r)

{

this.id=i;

this.radius=r;

}

void setID(double i)

{

this.id=i;

}

double getID()

{

return(this.id);

}

void setRadius(double r)

{

this.radius=r;

}

double getRadius()

{

return(this.radius);

}

double getArea()

{

area = Math.PI \* (radius \* radius);

return (area);

}

public String toString()

{

return (""+id+"\t"+radius+"\t"+getArea());

}

public void getData() throws Exception

{

DataInputStream in=new DataInputStream(System.in);

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

id=Double.parseDouble(in.readLine());

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

radius=Double.parseDouble(in.readLine());

}

}

class program\_34

{

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

{

DataInputStream in=new DataInputStream(System.in);

String filename="./file/file1.data";

int limit=0,option=0;

ArrayList<Circle> objects=new ArrayList<Circle>();

while(option!=5)

{

System.out.println("1. Add new record");

System.out.println("2. Display record(s)");

System.out.println("3. Search a record");

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

System.out.println("5. exit");

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

System.out.print("Enter Your Choice : ");

option=Integer.parseInt(in.readLine());

switch(option)

{

case 1:

System.out.print("\nEnter the record \n");

Circle s= new Circle();

s.getData();

objects.add(s);

writeObjectsToFile(filename,objects);

System.out.println("\n Record successfully saved..... \n ");

break;

case 2:

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

objects=readObjectsFromFile(filename);

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

System.out.println("id\tR\tarea");

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

for(Object object : objects)

{

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

}

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

break;

case 3:

boolean found\_flag=false;

double search;

System.out.print("\n Enter circle id to search a record : \n ");

search=Double.parseDouble(in.readLine());

objects=readObjectsFromFile(filename);

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

System.out.println("id\tR\tarea");

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

for(Object object : objects)

{

if(((Circle)object).getID()==search)

{

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

found\_flag=true;

break;

}

}

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

if(found\_flag==false)

{

System.out.print("Record not found....");

}

break;

case 4:

double delete;

boolean delete\_flag=false;

System.out.print("Enter a circle id to delete a record : ");

delete=Integer.parseInt(in.readLine());

objects=readObjectsFromFile(filename);

int delete\_index=0;

for(Object object : objects)

{

if(((Circle)object).getID()==delete)

{

delete\_flag=true;

break;

}

delete\_index++;

}

if(delete\_flag==true)

{

objects.remove(delete\_index);

writeObjectsToFile(filename,objects);

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

}

else

{

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

}

break;

case 5:

System.out.println("\n You are exited from program..\n ");

break;

default:

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

break;

}

}

}

private static void writeObjectsToFile(String filename,ArrayList<Circle> objects)

{

try

{

FileOutputStream fos=new FileOutputStream(filename);

ObjectOutputStream oos=new ObjectOutputStream(fos);

for(Object object:objects)

{

oos.writeObject(object);

}

oos.flush();

oos.close();

fos.close();

}

catch(Exception e)

{

e.printStackTrace();

}

}

private static ArrayList<Circle> readObjectsFromFile(String filename) throws Exception

{

ArrayList<Circle> objects=new ArrayList<Circle>();

FileInputStream fis=new FileInputStream(filename);

ObjectInputStream ois =new ObjectInputStream(fis);

while(true)

{

try

{

Circle object=(Circle)ois.readObject();

objects.add(object);

}

catch(Exception e)

{

break;

}

}

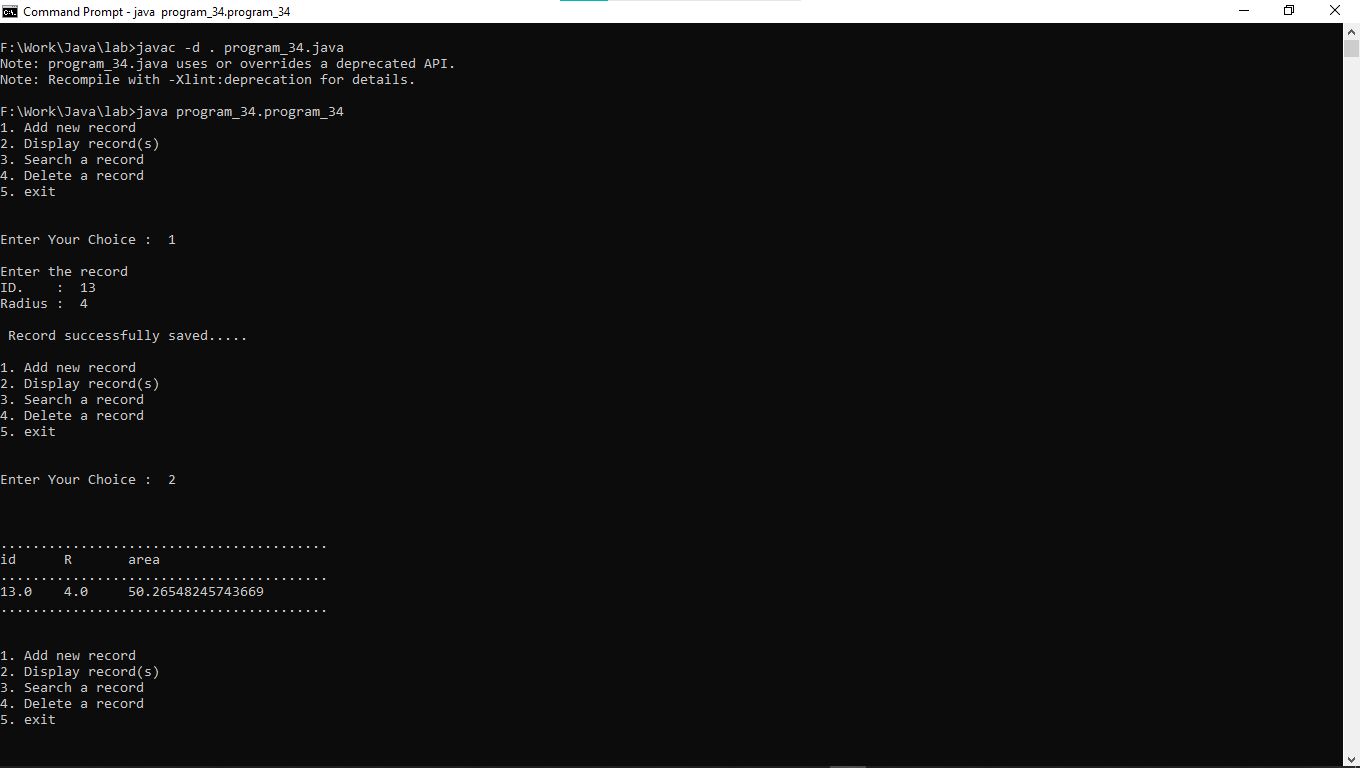
ois.close();

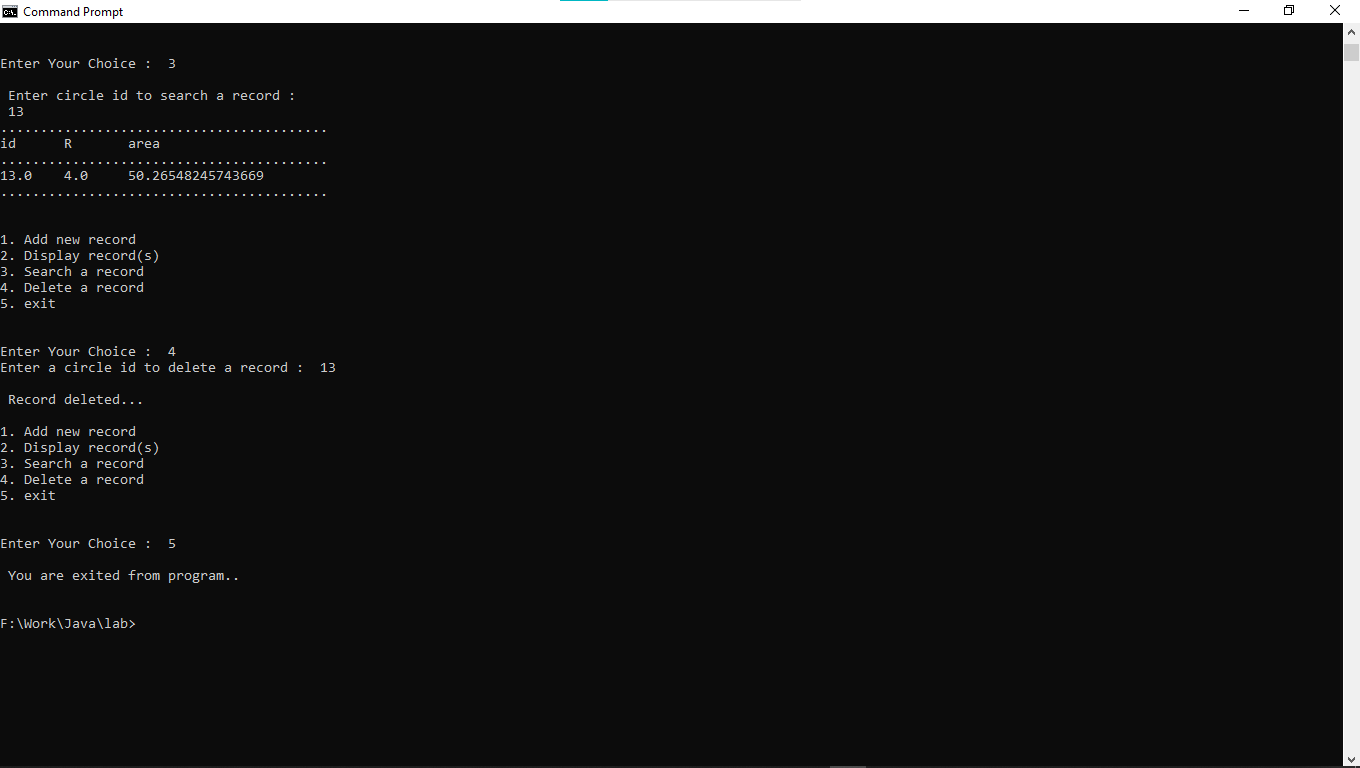
fis.close();

return objects;

}

}

**Output:-**

****

**35 Write a program for Java Generics class for 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. [desirable]**

package program\_35;

import java.util.\*;

class Student {

int rollno;

String name;

int age;

Student(int rollno, String name, int age) {

this.rollno = rollno;

this.name = name;

this.age = age;

}

}

class NameComparator implements Comparator<Student> {

public int compare(Student s1, Student s2) {

return s1.name.compareTo(s2.name);

}

}

class AgeComparator implements Comparator<Student> {

public int compare(Student s1, Student s2) {

if (s1.age == s2.age)

return 0;

else if (s1.age > s2.age)

return 1;

else

return -1;

}

}

public class program\_35 {

public static void main(String[] args) {

List<String> list = Arrays.asList("keyur", "parth", "jugal", "akash");

System.out.println("");

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

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

Collections.sort(list);

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

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

Collections.reverse(list);

System.out.println("sorting after Reversing: " + list);

System.out.println("\n SORTING CUSTOM TYPES ");

ArrayList<Integer> emp = new ArrayList<Integer>();

emp.add(45);

emp.add(56);

emp.add(30);

emp.add(25);

emp.add(26);

emp.add(34);

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

Collections.sort(emp);

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

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

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

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

al.add(new Student(101, "Vijay", 23));

al.add(new Student(106, "Ajay", 27));

al.add(new Student(105, "Jai", 21));

System.out.println("Sorting by Name");

Collections.sort(al, new NameComparator());

for (Student st : al) {

System.out.println(st.rollno + " " + st.name + " " + st.age);

}

System.out.println("");

System.out.println("Sorting by age");

Collections.sort(al, new AgeComparator());

for (Student st : al) {

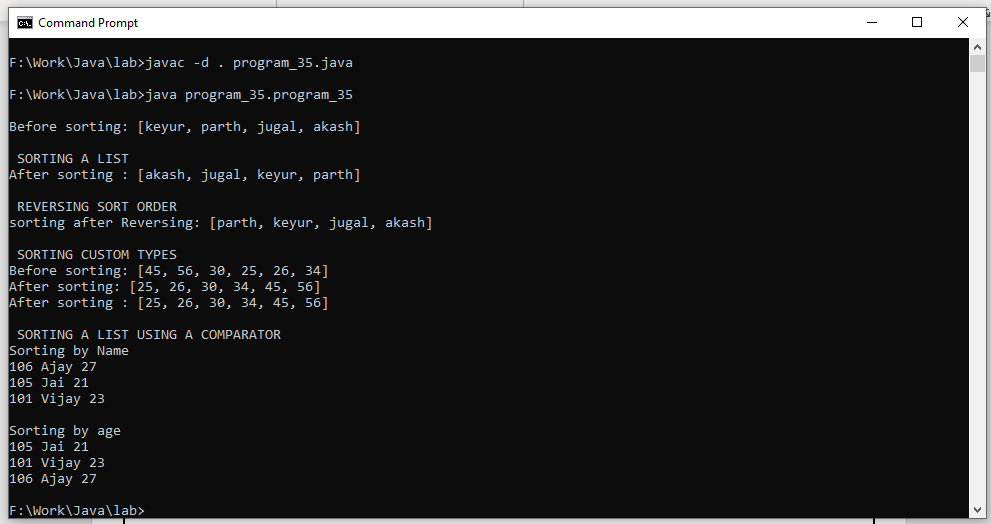
System.out.println(st.rollno + " " + st.name + " " + st.age);

}

}

}

**Output:-**



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

import java.io.\*;

public class program\_36 {

public static void main(String[] args) {

try

{

String check,marge;

//create first File

File f1=new File("./file/file.txt");

f1.createNewFile();

FileWriter fw=new FileWriter(f1);

BufferedWriter bf=new BufferedWriter(fw);

bf.write("H K M");

bf.flush();

//create Second file

File f2=new File("./file/file2.txt");

f2.createNewFile();

FileWriter fw1=new FileWriter(f2);

BufferedWriter fb1=new BufferedWriter(fw1);

// read first file

FileReader r1=new FileReader(f1);

BufferedReader b1=new BufferedReader(r1);

//assign first file all the record to the check variable

while((check=b1.readLine())!=null)

{

System.out.println("File1: "+String.valueOf(check));//read first file records

marge=check.replaceAll("H K", "Halpati Keyur");// modify file records

fb1.write(marge);

System.out.println("Modify Records: "+String.valueOf(marge));//print modify records

}

b1.close();

}

catch(IOException e)

{

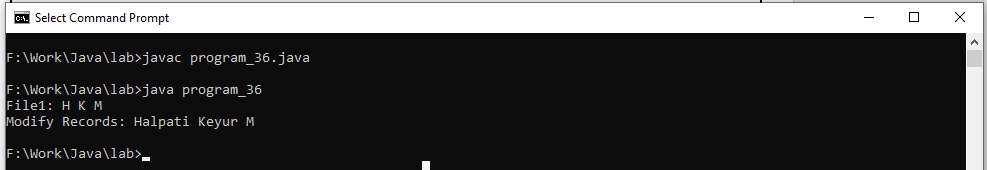
e.printStackTrace();

}

}

}

**Output:-**



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

import java.io.InputStreamReader;

import java.io.BufferedReader;

import java.io.IOException;

import java.util.Scanner;

import java.io.Console;

public class program\_37 {

public static void main(String[] args) {

try {

InputStreamReader in = new InputStreamReader(System.in);

BufferedReader read = new BufferedReader(in);

System.out.println("1. Scanner InputStream");

System.out.println("2. InputStreamReader");

System.out.println("3. Console InputStream");

System.out.println("4. Read InputStream");

System.out.println("Enter Your Choices: ");

int ch = Integer.parseInt(read.readLine());

switch (ch) {

case 1:

Scanner sc=new Scanner(System.in);

System.out.println("Enter Your Name: ");

String nm=sc.nextLine();

System.out.print("Name is "+nm);

break;

case 2:

System.out.println("Enter Your Name: ");

String inm=read.readLine();

System.out.print("Name is "+inm);

break;

case 3:

Console con= System.console();

String name=con.readLine("Enter Name\n");

System.out.println("Name is "+name);

System.out.println("Enter Your Age: ");

int age=Integer.parseInt(System.console().readLine());

System.out.print("Your Age: "+age);

break;

case 4:

System.out.println("Enter a Character:");

int inChar = System.in.read();

System.out.print("You entered ");

System.out.println(inChar);

break;

default:

break;

}

} catch (IOException e) {

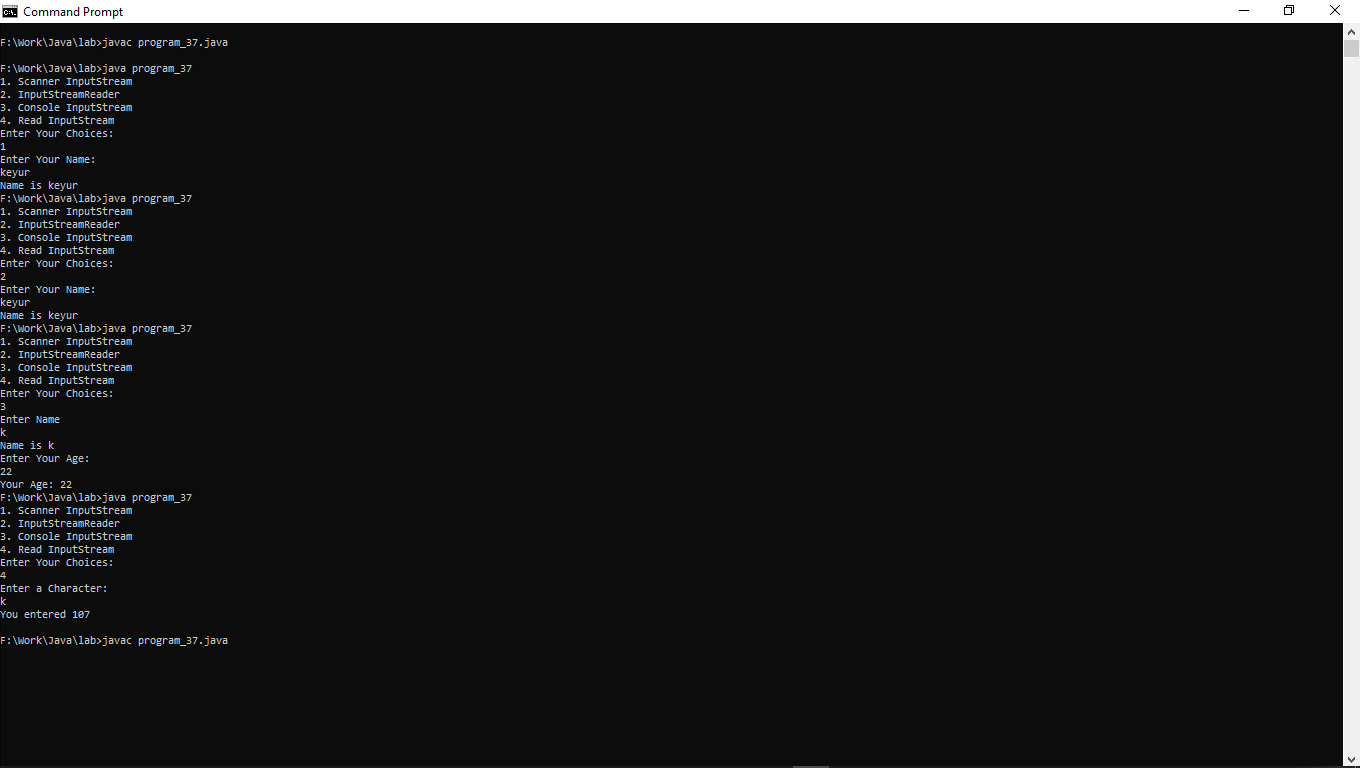
System.out.println(e);

}

}

}

**Output:-**



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

package program\_38;

import java.io.File;

public class program\_38 {

public static void main(String[] args) {

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

if(f.exists())

{

System.out.println("existence of a specified file");

}

else

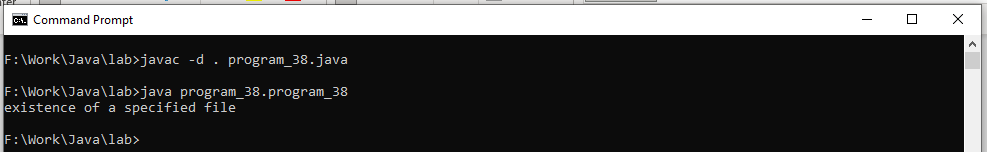
{

System.out.println("Not existence of a specified file");

}

}

}

****

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

package program\_39;

import java.io.File;

import java.io.IOException;

public class program\_39 {

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

File f=new File("./file/demo.txt");

boolean created=f.createNewFile();

if(created)

{

System.out.println("create a file to the specified location");

}

else

{

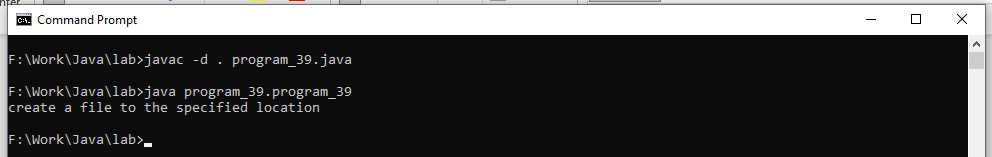
System.out.println("Not create a file to the specified location");

}

}

}

**Output:-**

****

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

package program\_40;

import java.io.FileInputStream;

import java.io.FileReader;

import java.io.IOException;

public class program\_40 {

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

FileInputStream file=new FileInputStream("./file/file2.txt");

int i=0;

System.out.print("File 1: ");

while((i=file.read())!=-1)

{

System.out.print((char)i);

}

file.close();

FileReader read=new FileReader("./file/file.txt");

int k=0;

System.out.println("");

System.out.print("File 2: ");

while((k=read.read())!=-1)

{

System.out.print((char)k);

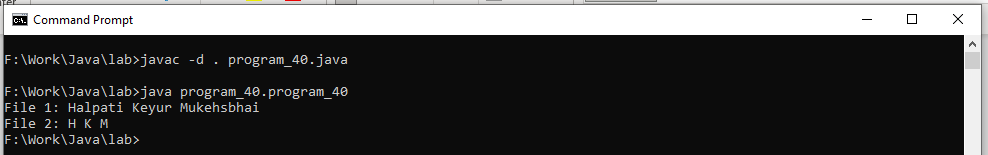
}

read.close();

}

}

**Output:-**



**41. 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.**

package program\_41;

import java.io.FileReader;

import java.io.IOException;

public class program\_41 {

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

FileReader file=new FileReader("./file/read.txt");

int i=0,c=0;

while((i=file.read())!=-1)

{

if(i=='\n')

{

c++;

}

System.out.print((char)i);

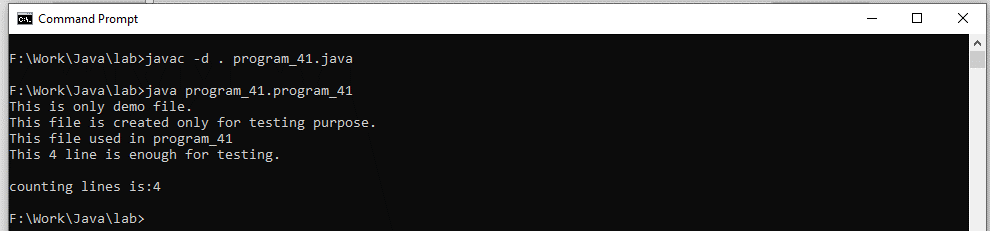
}

System.out.println("");

System.out.println("counting lines is:"+c);

}

}

**Output:-**

**42. 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).**

package program\_42;

import java.util.\*;

import java.util.function.\*;

class Counter {

static <T> int count(Collection<T> c, Predicate<T> p) {

int count = 0;

for (T e : c) {

if (p.test(e)) {

count++;

}

}

return count;

}

}

class Operation {

boolean EvenNumber(int num) {

return num % 2 == 0;

}

boolean OddNumber(int num) {

return num % 2 != 0;

}

boolean Palindrome(Integer num) {

int r, sum = 0, temp;

temp = num;

while (num > 0) {

r = num % 10;

sum = (sum \* 10) + r;

num = num / 10;

}

return (temp == sum);

}

}

public class program\_42 {

public static void main(String[] args) {

Collection<Integer> elements = Arrays.asList(1, 2, 3, 4, 5, 7, 8, 9, 10);

Operation ob = new Operation();

int count = -1;

count = Counter.count(elements, ob::OddNumber);

System.out.println("Number of odd : " + count);

count = Counter.count(elements, ob::EvenNumber);

System.out.println("Number of even : " + count);

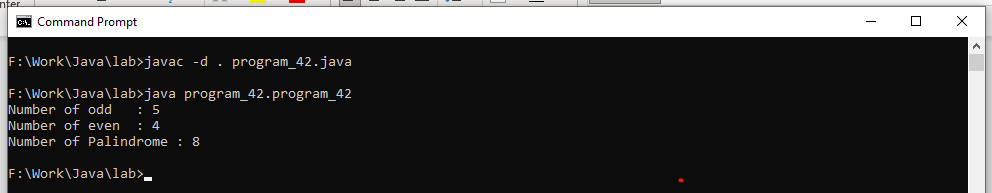
count = Counter.count(elements, ob::Palindrome);

System.out.println("Number of Palindrome : " + count);

}

}

**Output:-**



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

package program\_43;

class Swaping<T>

{

<T> void swap(T[] array, int l, int r)

{

T temp = array[r];

array[r] = array[l];

array[l] = temp;

}

<T> void Display(T[] array)

{

for(T a:array)

{

System.out.print(" "+a);

}

}

}

public class program\_43 {

public static void main(String[] args) {

Swaping<Integer> ob=new Swaping<Integer>();

Integer[] a={5,8,4,3};

System.out.println("Display Array");

ob.Display(a);

System.out.println("");

System.out.println("Display Swap Array");

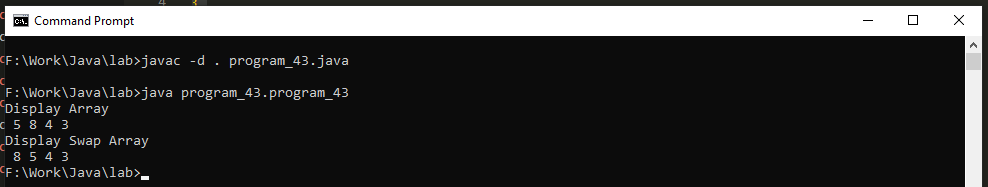
ob.swap(a, 0, 1);

ob.Display(a);

}

}

**Output:-**

****

**Thread Programming**

**1. Write a program to implement the concept of threading by extending “Thread” Class.**

class mythread extends Thread {

mythread(String name) {

super(name);

}

public void run() {

System.out.println("Start Name " + getName() + " Thread");

System.out.println("");

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(1000);

System.out.println("Thread " + getName() + " Count " + i);

}

} catch (InterruptedException e) {

System.out.println(e);

}

System.out.println("");

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

}

}

class program\_1 {

public static void main(String args[]) {

System.out.println("Thread Starting...");

mythread t = new mythread("K4");

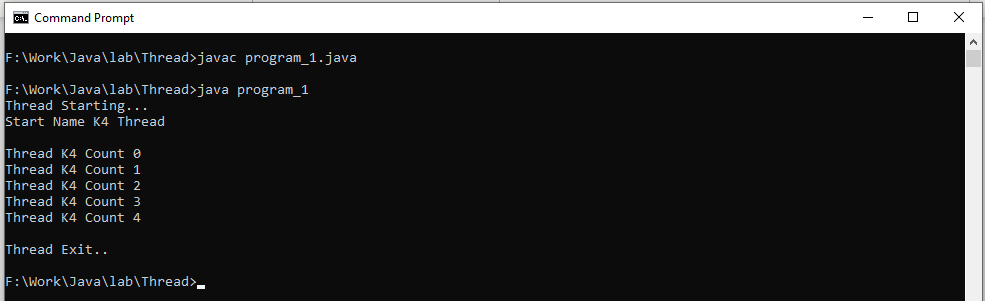
Thread my = new Thread(t);

my.start();

}

}

**Output:-**



**2. Write a program to implement the concept of threading by implementing “Runnable” Interface.**

class mythread implements Runnable {

String name;

mythread(String name) {

this.name = name;

}

public void run() {

System.out.println("Thread " + name + " Start");

System.out.println("");

try {

for (int i = 0; i < 5; i++) {

Thread.sleep(4000);

System.out.println("Thread " + name + " Count " + i);

}

} catch (InterruptedException e) {

System.out.println(e);

}

System.out.println("");

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

}

}

class program\_2 {

public static void main(String args[]) {

System.out.println("Thread Start...");

mythread t = new mythread("K1");

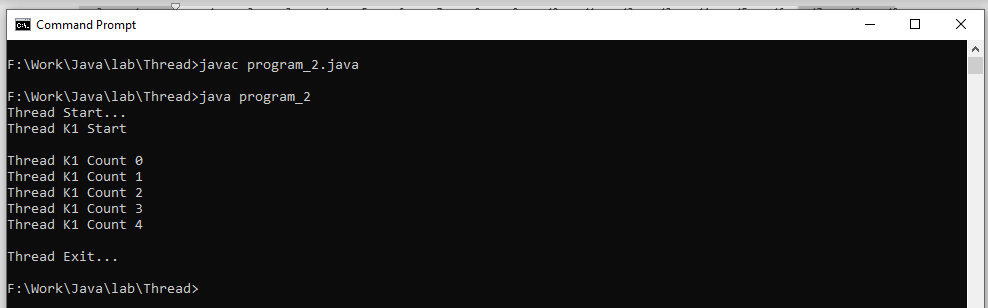
Thread th = new Thread(t);

th.start();

}

}

**Output:-**



**3. Write a program that executes two threads. One thread displays “Thread1” every 2,000 milliseconds, and the other displays “Thread2” every 4,000 milliseconds.**

class mythread extends Thread

{

mythread(String s1)

{

super(s1);

}

public void run()

{

System.out.println("Starting Thread...");

for(int i=0;i<5;i++)

{

System.out.println(Thread.currentThread().getName());

try

{

if(Thread.currentThread().getName()=="Thread1")

{

Thread.sleep(2000);

}

else

{

Thread.sleep(4000);

}

}

catch(InterruptedException e){

System.out.println(e);

}

}

}

}

class program\_3

{

public static void main(String args[])

{

mythread t1=new mythread("Thread1");

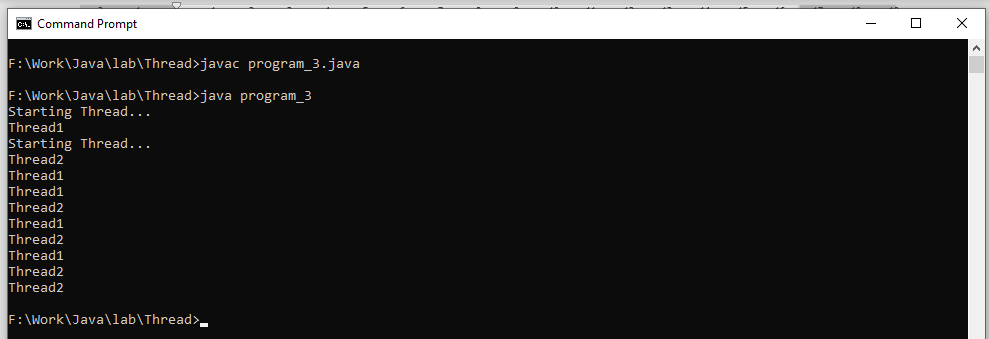
t1.start();

mythread t2=new mythread("Thread2");

t2.start();

}

}

**Output:-**

**4. Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 50.**

package program\_4;

class number

{

synchronized void numberN(int n)

{

for (int i = 1; i <= 50; i++) {

if(i%2==n)

{

System.out.println("" + Thread.currentThread().getName());

System.out.println(i);

}

}

}

}

class mythread1 implements Runnable

{

number n;

mythread1(number n)

{

this.n=n;

}

public void run() {

n.numberN(0);

}

}

class mythread2 implements Runnable

{

number n;

mythread2(number n)

{

this.n=n;

}

public void run() {

n.numberN(1);

}

}

public class program\_4 {

public static void main(String args[])

{

number n1=new number();

mythread1 t=new mythread1(n1);

mythread2 t1=new mythread2(n1);

Thread k=new Thread(t);

Thread k1=new Thread(t1);

k.start();

k1.start();

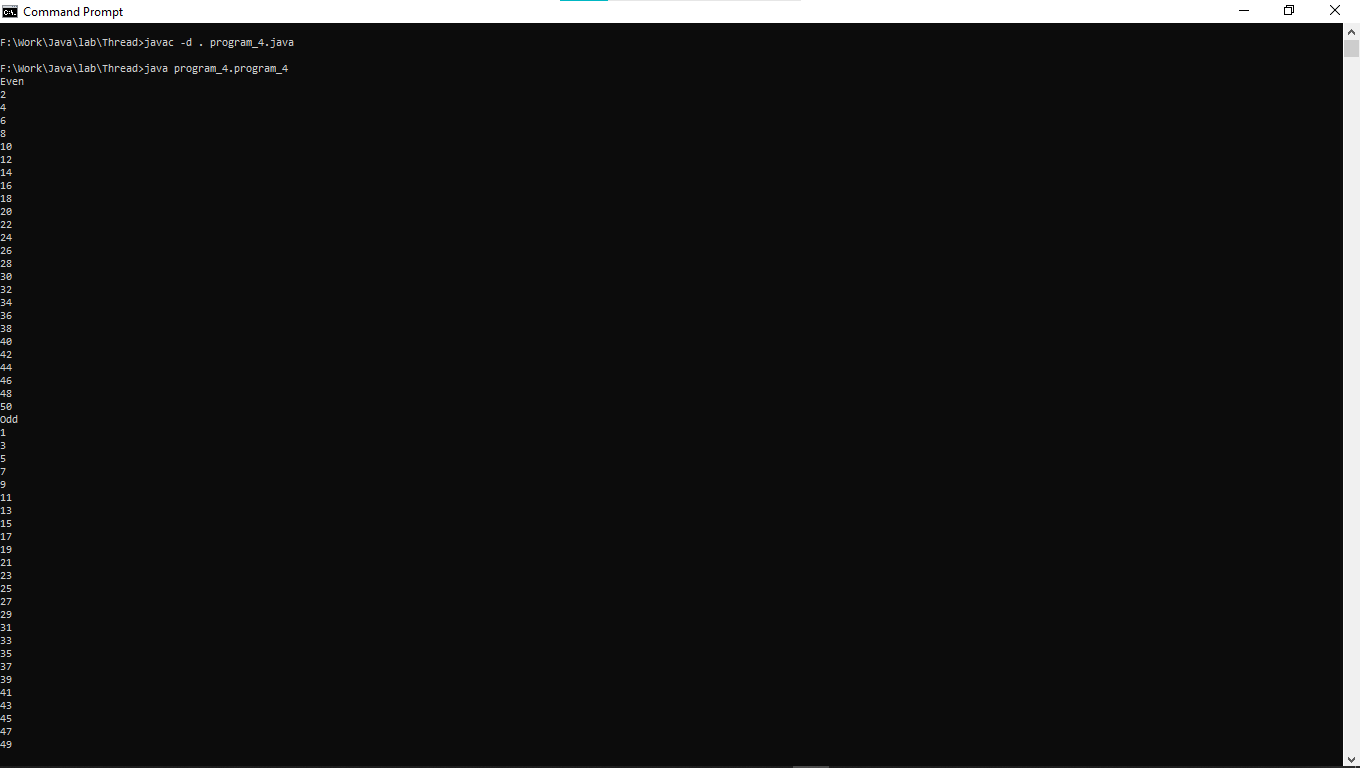
k.setName("Even");

k1.setName("Odd");

}

}

**Output:-**



**5. Write java program that create and runs following threads:**

**i) print "A" 20 times**

**ii) print "B" 30 times**

**iii) print "C" 15 times.**

package program\_5;

class ABC

{

synchronized void display(char c,int n)

{

System.out.println(" "+Thread.currentThread().getName());

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

{

System.out.println(c);

try

{

Thread.sleep(400);

}

catch(Exception e)

{

System.out.println("Intrrupted exception..");

}

}

}

}

class thread1 extends Thread

{

ABC a;

thread1(ABC a)

{

this.a=a;

}

public void run()

{

a.display('A',20);

}

}

class thread2 extends Thread

{

ABC b;

thread2(ABC b)

{

this.b=b;

}

public void run()

{

b.display('B',30);

}

}

class thread3 extends Thread

{

ABC c;

thread3(ABC c)

{

this.c=c;

}

public void run()

{

c.display('C',15);

}

}

class program\_5

{

public static void main(String args[])

{

ABC obj=new ABC();

thread1 t=new thread1(obj);

t.start();

thread2 t1=new thread2(obj);

t1.start();

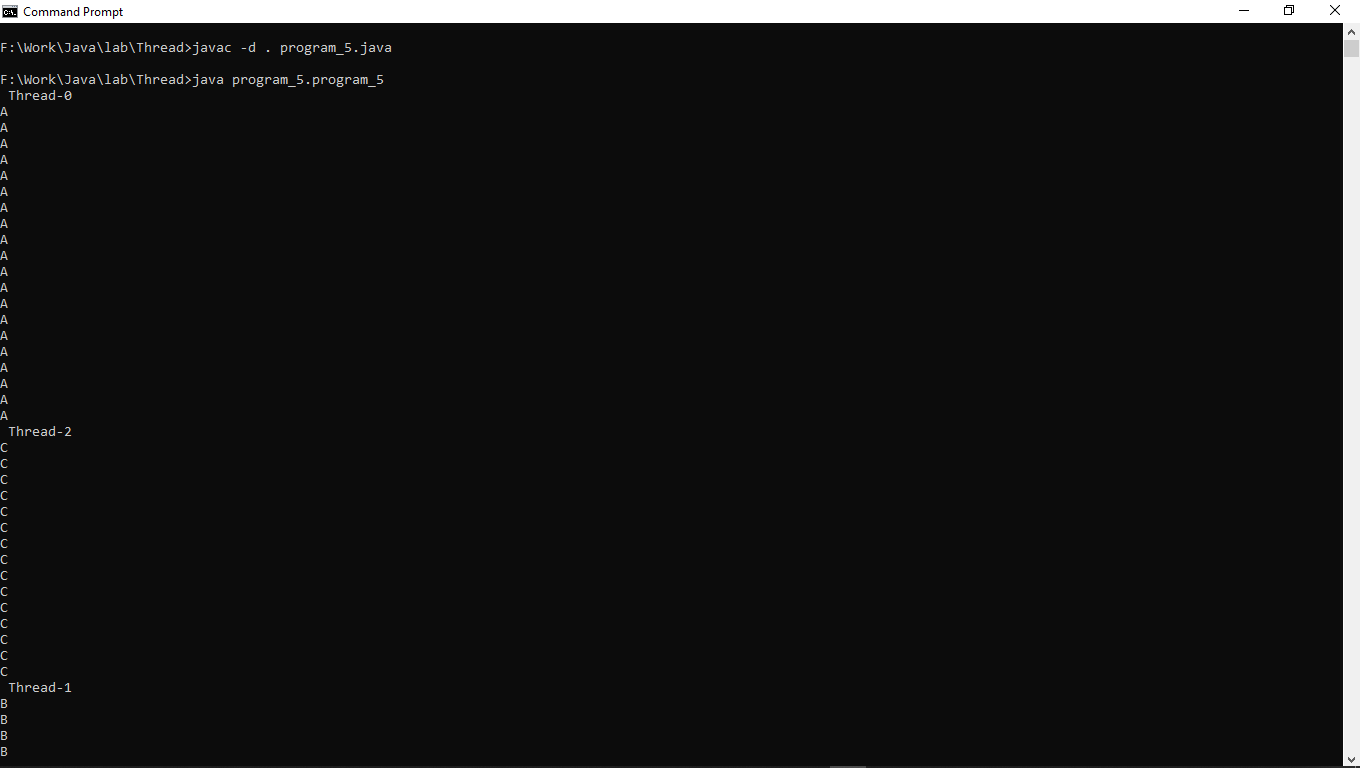
thread3 t2=new thread3(obj);

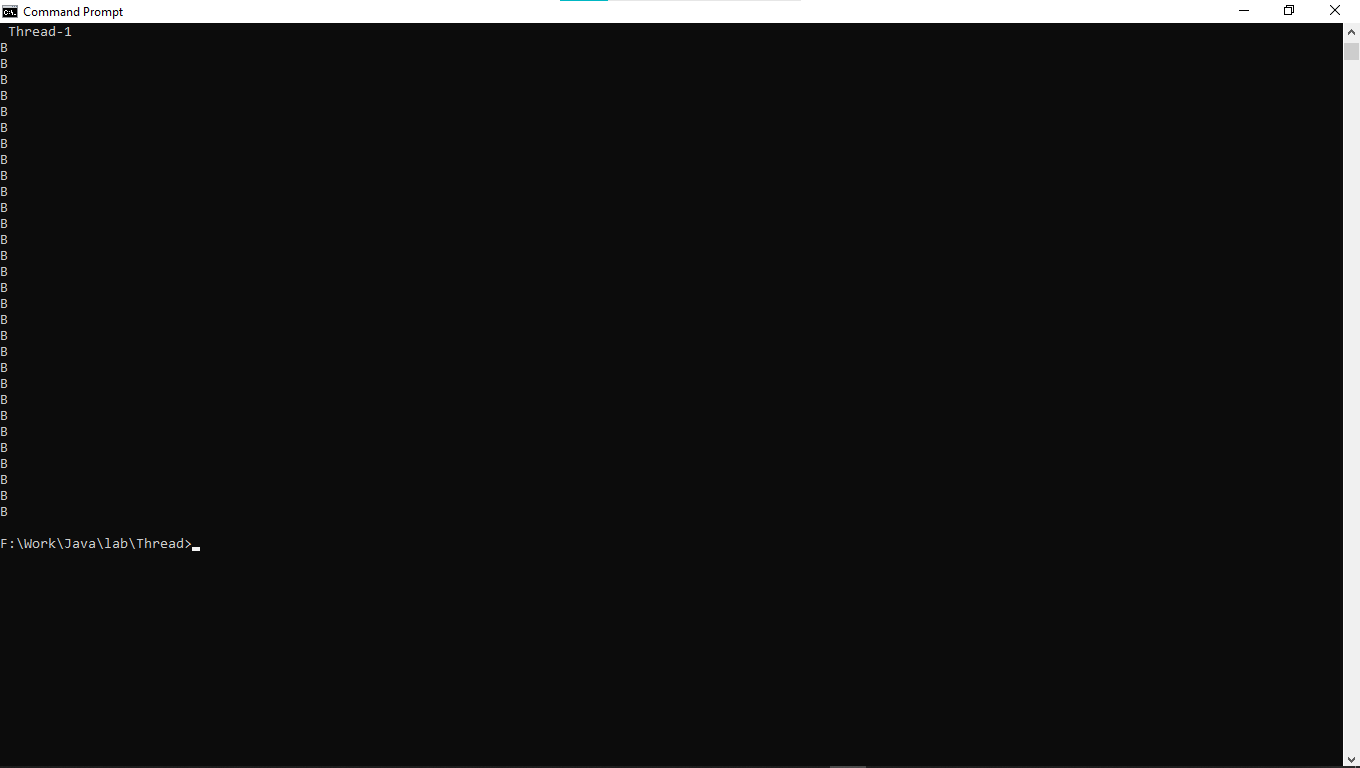
t2.start();

}

}

**Output:-**





**6. Write a program in Java to demonstrate use of synchronization of threads when multiple threads are trying to update common variable for “Account” class.**

package program\_6;

class Account

{

int bal,wid,newval;

synchronized void operation(int bal,int wid)

{

this.bal=bal;

this.wid=wid;

System.out.println(""+Thread.currentThread().getName());

try{

System.out.println("your Accounr balance is "+bal);

newval=bal-wid;

System.out.println("withdrow value is "+wid);

System.out.println("your current balance is "+newval);

Thread.sleep(4000);

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

System.out.println("");

}

}

}

class savingAccount extends Thread

{

Account s;

savingAccount(String s1,Account s)

{

super(s1);

this.s=s;

}

public void run()

{

s.operation(10000,2000);

}

}

class currentAccount extends Thread

{

Account c;

currentAccount(String s1,Account c)

{

super(s1);

this.c=c;

}

public void run()

{

c.operation(10000,2000);

}

}

class program\_6

{

public static void main(String agrg[])

{

Account p=new Account();

savingAccount s1=new savingAccount("SavingAccount",p);

s1.start();

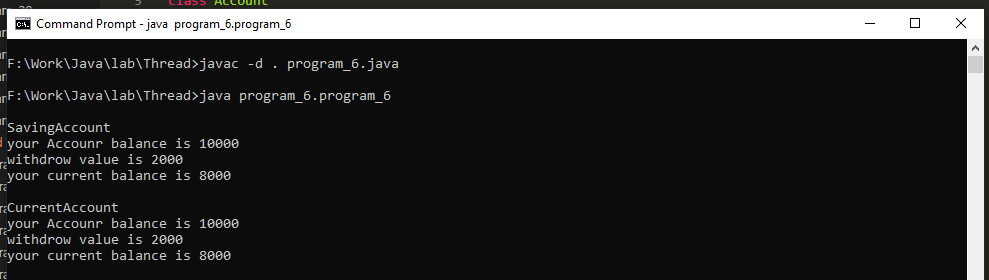
currentAccount s2=new currentAccount("CurrentAccount",p);

System.out.println("");

s2.start();

}

}

**Output:-**

**The collection Framework**

**1. Develop a program to create Linked List for “Student” class objects references. “Student” class has std\_id, std\_name, Array of marks, total\_ marks. Calculate total\_marks for all students of Linked List. Display Linked List and also display a particular student based on student name as a command line argument.**

package program\_5;

import java.io.IOException;

import java.util.\*;

class student {

int std\_id,total\_marks;

ArrayList<Integer> marks;

String std\_name;

public student(int std\_id, String std\_name, ArrayList<Integer> m) {

this.std\_id=std\_id;

this.std\_name=std\_name;

this.marks=m;

}

}

public class program\_5 {

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

LinkedList<student> list = new LinkedList<student>();

ArrayList<Integer> m=new ArrayList<Integer>();

ArrayList<Integer> n=new ArrayList<Integer>();

ArrayList<Integer> p=new ArrayList<Integer>();

m.add(70);

m.add(80);

m.add(60);

m.add(70);

n.add(70);

n.add(60);

n.add(90);

n.add(50);

p.add(50);

p.add(60);

p.add(70);

p.add(80);

student b1 = new student(1,"Keyur",m);

student b2 = new student(2,"Parth",n);

student b3 = new student(3,"Akash",p);

list.add(b1);

list.add(b2);

list.add(b3);

System.out.println("std\_id std\_name\tmarks \t\ttotal\_ marks");

for (student b : list) {

for (int i: b.marks)

{

b.total\_marks += i;

}

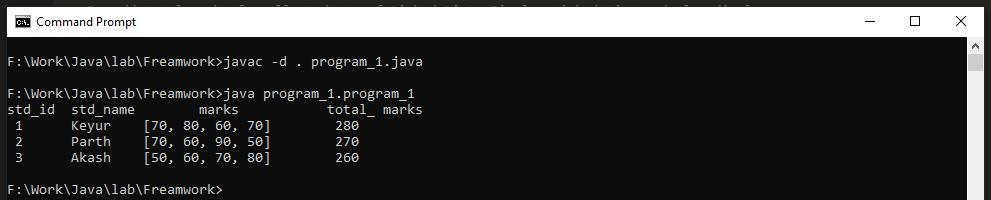
System.out.println(" "+b.std\_id + "\t" + b.std\_name + "\t " + b.marks+"\t "+b.total\_marks);

}

}

}

**Output:-**



**2. Develop a program to create Array List for “Employee” class objects references. Employee class has emp\_code, emp\_name, basic\_sal, gross\_ sal. Calculate gross\_sal for all employees of Array List. Display Array List and also insert an employee object reference in a particular position (input) in Array List. Gross\_sal=basic\_sal+20% of basic\_sal (MA)+30% of basic\_sal(HRA).**

package program\_2;

import java.util.ArrayList;

import java.util.Iterator;

class Employee {

int emp\_code, basic\_sal, gross\_sal;

String emp\_name;

Employee(String emp\_name, int emp\_code, int basic\_sal) {

this.emp\_code = emp\_code;

this.basic\_sal = basic\_sal;

this.emp\_name = emp\_name;

}

}

public class program\_2 {

public static void main(String[] args) {

Employee e1 = new Employee("keyur", 13, 10000);

Employee e2 = new Employee("Mayuri", 12, 30000);

Employee e3 = new Employee("Ketan", 15, 12000);

Employee e4 = new Employee("Yogita", 8, 11000);

ArrayList<Employee> emp = new ArrayList<Employee>();

emp.add(e1);

emp.add(e2);

emp.add(e3);

emp.add(1, e4);

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

System.out.println("emp\_code emp\_name basic\_sal gross\_ sal");

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

for (Employee ep : emp) {

ep.gross\_sal = ep.basic\_sal + 20 % ep.basic\_sal \* (1250) + 30 % ep.basic\_sal \* (15000);

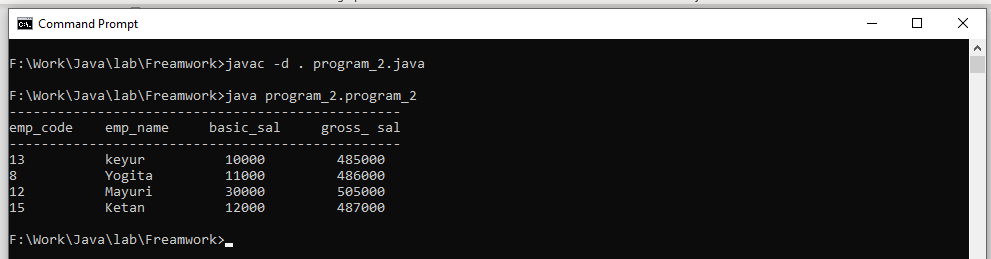
System.out.println(ep.emp\_code + "\t " + ep.emp\_name + "\t " + ep.basic\_sal + "\t " + ep.gross\_sal);

}

}

}

**Output:-**



**3. Develop a program to create Hash Map for “Customer” class objects references. Customer class has Bill\_no, cust\_mobile\_no, Array of item\_name, Array of item\_unit\_price, Array of item\_count, total\_price. Calculate total\_price for all customers of Hash Map. Display Hash Map and also search particular customer’s bill based on customer mobile no. Key is mobile no.**

package program\_3;

import java.util.\*;

class customer

{

int Bill\_no,cust\_mobile\_no,item\_unit\_price,item\_count,total\_price;

String item\_name;

customer(int Bill\_no,int cust\_mobile\_no,String item\_name,int item\_unit\_price,int item\_count)

{

this.Bill\_no=Bill\_no;

this.cust\_mobile\_no=cust\_mobile\_no;

this.item\_name=item\_name;

this.item\_unit\_price=item\_unit\_price;

this.item\_count=item\_count;

}

}

public class program\_3 {

public static void main(String[] args) {

HashMap<Integer,customer> map=new HashMap<Integer,customer>();

customer c1=new customer(10, 123456, "iphone", 12000, 2);

customer c2=new customer(11, 123457, "Camera", 11000, 3);

customer c3=new customer(12, 123458, "smartp", 11000, 2);

customer c4=new customer(13, 123459, "SofaSet", 10000, 4);

map.put(c1.cust\_mobile\_no, c1);

map.put(c2.cust\_mobile\_no, c2);

map.put(c3.cust\_mobile\_no, c3);

map.put(c4.cust\_mobile\_no, c4);

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

System.out.println("Bill\_no mobile\_no item\_name price item\_count total\_price");

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

for(Map.Entry<Integer, customer> entry:map.entrySet()){

int key=entry.getKey();

customer c=entry.getValue();

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

// System.out.println(key+" Details:");

c.total\_price=c.item\_unit\_price\*c.item\_count;

System.out.println(" "+c.Bill\_no+" "+c.cust\_mobile\_no+" "+c.item\_name+" "+c.item\_unit\_price+" "+c.item\_count+" "+c.total\_price);

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

}

System.out.println("");

System.out.println("searching particular customer's bill based on customer mobile no");

Scanner in=new Scanner(System.in);

int key=in.nextInt();

System.out.println("");

Boolean flag=map.containsKey(key);

if(flag==true)

{

System.out.println("customer's bill is Details: ");

customer c=map.get(key);

c.total\_price=c.item\_unit\_price\*c.item\_count;

System.out.println(c.Bill\_no+" "+c.cust\_mobile\_no+" "+c.item\_name+" "+c.item\_unit\_price+" "+c.item\_count+" "+c.total\_price);

System.out.println("");

}

else

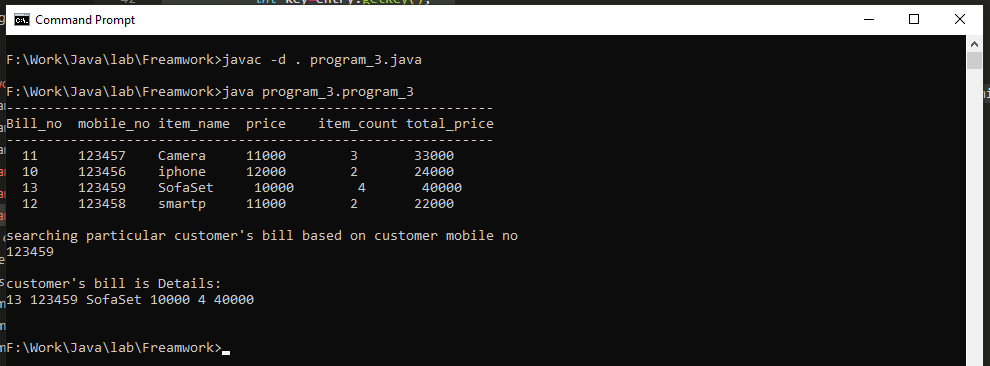
{

System.out.println("Not Existed customer's bill");

}

}

}

**Output:-**

**4. Sort “Student” Linked List (mentioned in Q:1) based on std\_name using “Comparator” interface.**

package program\_4;

import java.util.\*;

class Student {

int rollno;

String name;

int age;

Student(int rollno, String name, int age) {

this.rollno = rollno;

this.name = name;

this.age = age;

}

}

class NameComparator implements Comparator {

public int compare(Object o1, Object o2) {

Student s1 = (Student) o1;

Student s2 = (Student) o2;

return s1.name.compareTo(s2.name);

}

}

public class program\_4 {

public static void main(String args[]) {

ArrayList al = new ArrayList();

al.add(new Student(1, "Keyur", 23));

al.add(new Student(2, "Jugal", 24));

al.add(new Student(3, "Parth", 25));

al.add(new Student(4, "Rahul", 20));

al.add(new Student(5, "Akash", 22));

System.out.println("Sorting by Name");

System.out.println("");

System.out.println("ID NAME AGE");

Collections.sort(al, new NameComparator());

Iterator itr = al.iterator();

while (itr.hasNext()) {

Student st = (Student) itr.next();

System.out.println("" + st.rollno + " " + st.name + " " + st.age);

}

}

}

**Output:-**