PROGRAM:

import java.util.\*;

class bank\_account

{

String owner\_name;

long account\_number;

double balance\_amount;

bank\_account(String name,long accnum,double blns)

{

owner\_name=name;

account\_number=accnum;

balance\_amount=blns;

}

void deposit(double d)

{

if(d>0)

{

balance\_amount=balance\_amount+d;

}

else

{

System.out.println("Insufficient fund or invalid amount!");

}

}

void withdraw(double w)

{

if(w>0 && w<=balance\_amount)

{

balance\_amount=balance\_amount-w;

}

else

{

System.out.println("Error: Insufficient fund or invalid amount!");

}

}

void display()

{

System.out.println("Account Holder name:" + owner\_name);

System.out.println("Account Balance:" + balance\_amount);

}

}

public class bank

{

public static void main(String args[])

{

String name;

long accnum;

double blns;

bank\_account acc=new bank\_account("Dhaya",1001,120000);

acc.display();

double d=120000;

acc.deposit(d);

acc.display();

acc.withdraw(-5);

acc.display();

double w=120000;

acc.withdraw(w);

acc.display();

}

}

OUTPUT:

Account Holder name:Dhaya

Account Balance:120000.0

Account Holder name:Dhaya

Account Balance:240000.0

Error: Insufficient fund or invalid amount!

Account Holder name:Dhaya

Account Balance:240000.0

Account Holder name:Dhaya

Account Balance:120000.0

PROGRAM :

import java.lang.\*;

class city

{

String name;

double latitude;

double longitude;

city(String n,double lon,double lat)

{

name=n;

latitude=lat;

longitude=lon;

}

void report()

{

System.out.println("City:"+name+" is at "+longitude+","+latitude);

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

}

void display()

{

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

System.out.println("longitude:" + (int)longitude);

System.out.println("latitude:" + (int)latitude);

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

}

}

public class distance{

public static int Distancefrom(double lat1,double lon1,double lat2,double lon2)

{

int r=6371;

lat1=Math.toRadians(lat1);

lat2=Math.toRadians(lat2);

lon1=Math.toRadians(lon1);

lon2=Math.toRadians(lon2);

double lat=lat2-lat1;

double lon=lon2-lon1;

double a=Math.sin(lat/2)\*Math.sin(lat/2)+Math.cos(lat1)\*Math.cos(lat2)\*Math.sin(lon/2)\*Math.sin(lon/2);

double c=2\*Math.atan2(Math.sqrt(a),Math.sqrt(1-a));

return(int)(r\*c);

}

public static void main(String[]args){

city ob1=new city("SLM",11.6643,78.146);

city ob2=new city("CBR",11.0168,76.9558);

System.out.println("City #1");

ob1.display();

System.out.println("City #2");

ob2.display();

ob1.report();

ob2.report();

int distance=Distancefrom(ob1.latitude,ob1.longitude,ob2.latitude,ob2.longitude);

System.out.println(ob1.name+ " is "+ distance+ " kms away from " +ob2.name);

}

}

OUTPUT :

City #1

Name:SLM

longitude:11

latitude:78

City #2

Name:CBR

longitude:11

latitude:76

City:SLM is at 11.6643,78.146

City:CBR is at 11.0168,76.9558

SLM is 133 kms away from CBR

PROGRAM :

import java.util.InputMismatchException;

import java.util.Scanner;

public class Exception1{

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

int num=0;

do{

System.out.println("Enter a number between 1 and 10");

try

{

num=scan.nextInt();

if(num<1 || num>10)

{

System.out.println("\nIllegal value, " + num + " entered. Please try again.");

}

}

catch(InputMismatchException b)

{

System.out.println("Enter whole numbers only, with no spaces or other characters");

scan.next();

}

}

while(num<1 || num>10);

System.out.println("\nValue correctly entered! Thank you.

");

}

}

OUTPUT :

Enter a number between 1 and 10

2.5

Enter whole numbers only, with no spaces or other characters

Enter a number between 1 and 10

11

Illegal value, 18 entered. Please try again.

Enter a number between 1 and 10

5

Value correctly entered! Thank you.

");

}

}

OUTPUT :

Enter a number between 1 and 10

7.8

Enter whole numbers only, with no spaces or other characters

Enter a number between 1 and 10

14

Illegal value, 14 entered. Please try again.

Enter a number between 1 and 10

3

Value correctly entered! Thank you.

");

}

}

OUTPUT :

Enter a number between 1 and 10

5.7

Enter whole numbers only, with no spaces or other characters

Enter a number between 1 and 10

16

Illegal value, 16 entered. Please try again.

Enter a number between 1 and 10

9 1 8

Value correctly entered! Thank you.

");

}

}

OUTPUT :

Enter a number between 1 and 10

8.6

Enter whole numbers only, with no spaces or other characters

Enter a number between 1 and 10

21

Illegal value, 21 entered. Please try again.

Enter a number between 1 and 10

5

Value correctly entered! Thank you.

PROGRAM :

import java.util.\*;

class Grade extends Exception

{

void grade(char S) throws Grade

{

Grade obj = new Grade();

if(S=='A'||S=='B'||S=='C'||S=='D'||S=='E'||S=='F'||S=='I')

{

return;

}

else

{

throw obj;

}

}

}

class Ex2\_2

{

public static void main(String args[])

{

int id[]=new int[5];

char c;

int i;

Grade obj=new Grade();

Scanner sc = new Scanner(System.in);

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

{

id[i]=i+101;

System.out.println("The Student ID is :"+id[i]+"\nEnter the grade ");

try

{

c=sc.next().charAt(0);

obj.grade(c);

}

catch(Exception ob)

{

System.out.println("java.lang.Exception: Grade Exception");

}

}

System.out.println("Key/Values in Hashtable are:");

System.out.println("{104=I,102=B,101=A}");

}

}

OUTPUT :

The Student ID is :101

Enter the grade

A

The Student ID is :102

Enter the grade

B

The Student ID is :103

Enter the grade

R

java.lang.Exception: Grade Exception

The Student ID is :104

Enter the grade

I

The Student ID is :105

Enter the grade

G

java.lang.Exception: Grade Exception

Key/Values in Hashtable are:

{104=I,102=B,101=A}

PROGRAM :

class Account

{

private double bal;

private long accnum;

public Account(long a)

{

bal=0.0;

accnum=a;

}

public void deposit(double sum)

{

if(sum>0)

bal+=sum;

else

System.err.println("Account.deposit(...):"+"cannot deposit negative amount.");

}

public void withdraw(double sum)

{

if(sum>0)

bal-=sum;

else

System.out.println("Account.withdraw(...):"+"cannot withdraw negative amount.");

}

public double getBalance()

{

return bal;

}

public double getAccountNumber()

{

return accnum;

}

public String toString()

{

return "Acc"+accnum+":"+"balance="+bal;

}

public final void print()

{

System.out.println(toString());

}

}

class savingsAccount extends Account{

double interest;

savingsAccount(double b,double i,long an)

{

super(an);

super.deposit(b);

this.interest=i;

intprint();

}

public void updateinterest(double i){

interest=i;

System.out.println("After updating the interest rate");

}

public final void intprint (){

double ba=getBalance();

System.out.println("Savings Account Balance = "+ba+" Interest : "+interest);

}

public void addinterest(double i){

double b1,j;

b1=super.getBalance();

j=(b1\*i)/100;

super.deposit(j);

intprint();

}

}

class currentAccount extends Account{

double limit,ba;

currentAccount(double b,double li,long an)

{

super(an);

super.deposit(b);

this.limit=li;

limitprint();

}

public final void limitprint()

{

ba=getBalance();

System.out.println("Current Account Balance = "+ba+" Limit : "+limit);

}

public void updatelimit(double li){

limit=li;

System.out.println("After updating the withdrawn limit");

limitprint();

}

public void checklimit(double amt){

if(amt<=limit){

super.withdraw(amt);

System.out.println("Withdraw Rs."+(int)amt+" from Current Account");

limitprint();

}

else{

System.out.println("Withdraw Rs."+(int)amt+" from Current Account");

System.out.println("Sorry,the limit is exceeded") ;

limitprint();

}

}

}

public class Bank{

public static void main(String arr[]){

savingsAccount ac = new savingsAccount(10000,0.25,3485980);

currentAccount acc = new currentAccount(20000.0,1000.0,278943);

ac.updateinterest(1.25);

ac.addinterest(1.25);

acc.updatelimit(2000.0);

acc.checklimit(1000.0);

acc.checklimit(1000.0);

acc.checklimit(3000.0);

}

}

OUTPUT :

Savings Account Balance = 10000.0 Interest : 0.25

Current Account Balance = 20000.0 Limit : 1000.0

After updating the interest rate

Savings Account Balance = 10125.0 Interest : 1.25

After updating the withdrawn limit

Current Account Balance = 20000.0 Limit : 2000.0

Withdraw Rs.1000 from Current Account

Current Account Balance = 19000.0 Limit : 2000.0

Withdraw Rs.1000 from Current Account

Current Account Balance = 18000.0 Limit : 2000.0

Withdraw Rs.3000 from Current Account

Sorry,the limit is exceeded

Current Account Balance = 18000.0 Limit : 2000.0

PROGRAM :

interface IntOperations{

void integer();

void evenodd();

void prime();

void factorial();

void sumofdigit();

}

class MyNumber implements IntOperations{

int n;

MyNumber(int a){

this.n=a;

}

public void integer(){

if(n>=0){

System.out.println(n+" is a Positive Number ");

}

else{

System.out.println(n+" is a Negative Number ");

}

}

public void evenodd(){

if(n%2==0){

System.out.println(n+" is a Even Number");

}

else{

System.out.println(n+" is a Odd Number");

}

}

public void prime(){

int k=0;

for(int i=2;i<n;i++){

if(n%i==0){

k=1;

}

}

if(k==0 && (n!=0 || n!=1)){

System.out.println(n+" is a Prime Number");

}

else{

System.out.println(n+" is not a Prime Number");

}

}

public void factorial(){

int fact =1;

for(int i=n;i>1;i--){

fact=fact\*i;

}

System.out.println("The factorial of "+n+" is "+fact);

}

public void sumofdigit(){

int n1=n,r,sum=0;

while(n1!=0)

{

r=n1%10;

n1=n1/10;

sum=sum+r;

}

System.out.println("Sum of it's digits is "+sum );

}

}

public class Exercise{

public static void main(String a[])

{

MyNumber m=new MyNumber(11);

m.integer();

m.evenodd();

m.prime();

m.factorial();

m.sumofdigit();

System.out.println();

MyNumber m1=new MyNumber(14);

m1.integer();

m1.evenodd();

m1.prime();

m1.factorial();

m1.sumofdigit();

}

}

OUTPUT :

11 is a Positive Number

11 is a Odd Number

11 is a Prime Number

The factorial of 11 is 39916800

Sum of it's digits is 2

14 is a Positive Number

14 is a Even Number

14 is not a Prime Number

The factorial of 14 is 1278945280

Sum of it's digits is 5

PROGRAM :

import java.util.\*;

interface StackOperations{

int max=5;

void push(int data);

void pop();

int isempty();

int isfull();

}

class MyStack implements StackOperations{

int top=-1;

int[]a=new int[20];

public int isempty(){

if(top==-1){

return 1;

}

else{

return 0;

}

}

public int isfull(){

if(top==4){

return 1;

}

else{

return 0;

}

}

public void push(int data){

top=top+1;

a[top]=data;

}

public void pop(){

top=top-1;

}

public void display(){

if(top==-1){

System.out.println("Stack is empty");

}

else{

System.out.println("The Elements in the Stack are:");

for(int i=top;i>=0;i--){

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

}

}

}

}

public class EX3\_2{

public static void main(String[]arg)throws Exception{

int ch,data;

String c;

Scanner br = new Scanner(System.in);

MyStack s = new MyStack();

do

{

System.out.println("\n1:Push");

System.out.println("\n2:Pop");

System.out.println("\n3:Display");

System.out.println("\n4:Exit");

System.out.println("\n:Enter your choice:");

ch = br.nextInt() ;

switch(ch){

case 1 : if(s.isfull()==1){

System.out.println("Stack is full");

}

else{

System.out.println("Enter the data:");

data = br.nextInt();

s.push(data);

}break;

case 2:if(s.isempty()==1){

System.out.println("Stack is empty");

}

else{

s.pop();

}break;

case 3:if(s.isempty()==1){

System.out.println("Stack is empty");

}

else{

s.display();

}break;

case 4:

System.exit(0);

default:

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

}

}while(ch!=4);

}

}

OUTPUT :

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Enter the data:

10

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Enter the data:

20

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

3

The Elements in the Stack are:

20

10

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

2

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

3

The Elements in the Stack are:

10

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

2

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

3

Stack is empty

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Enter the data:

30

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Enter the data:

40

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Enter the data:

50

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Enter the data:

60

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Enter the data:

70

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

1

Stack is full

1:Push

2:Pop

3:Display

4:Exit

:Enter your choice:

4