**Experiment No 1**

**Problem Statement:**

Create a class called Employee that includes three pieces of information as instance variables- first name, a last name and a monthly salary. Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.

**//Source Code**

**public** **class** Employee {

**public** String Fname,Lname;

**public** **double** salary;

Employee()

{

Fname=**null**;

Lname=**null**;

salary=0.0;

}

**public** **void** setFname(String fname) {

Fname = fname;

}

**public** **void** setLname(String lname) {

Lname = lname;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

**if**(salary<0)

{

salary=0.0;

}

}

**public** String getFname() {

**return** Fname;

}

**public** String getLname() {

**return** Lname;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** raiseSalary()

{

salary=salary+salary/10;

}

}

**public** **class** Employee\_Test {

**public** **static** **void** main(String[] args)

{

Employee e1=**new** Employee();

Employee e2=**new** Employee();

System.***out***.println("EMPLOYEE DETAILS ");

e1.setFname("Alisha");

e1.setLname("Mugale");

e1.setSalary(1000);

System.***out***.println("\nFirst Name: "+e1.Fname);

System.***out***.println("Last Name: "+e1.Lname);

System.***out***.println("Salary: "+e1.salary);

e2.setFname("Aditya");

e2.setLname("Patil");

e2.setSalary(2000);

System.***out***.println("\nFirst Name: "+e2.Fname);

System.***out***.println("Last Name: "+e2.Lname);

System.***out***.println("Salary: "+e2.salary);

System.***out***.println("\nEMPLOYEE DETAILS AFTER INCREMENT SALARY BY 10%");

e1.setFname("Alisha");

e1.setLname("Mugale");

e1.setSalary(1000);

e1.raiseSalary();

System.***out***.println("\nFirst Name: "+e1.Fname);

System.***out***.println("Last Name: "+e1.Lname);

System.***out***.println("Salary: "+e1.salary);

e1.setFname("Alisha");

e1.setLname("Mugale");

e1.setSalary(1000);

e2.raiseSalary();

System.***out***.println("\nFirst Name: "+e2.Fname);

System.***out***.println("Last Name: "+e2.Lname);

System.***out***.println("Salary: "+e2.salary);

}

}

**Output:**

EMPLOYEE DETAILS BEFORE SALARY INCREMENT

First Name: Alisha

Last Name: Mugale

Salary: 1000.0

First Name: Aditya

Last Name: Patil

Salary: 2000.0

EMPLOYEE DETAILS AFTER INCREMENT SALARY BY 10%

First Name: Alisha

Last Name: Mugale

Salary: 1100.0

First Name: Aditya

Last Name: Patil

Salary: 2200.0

**Experiment No 2**

**Problem Statement:**

Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value

Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs 2000.00 and Rs 3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.

**//Source Code**

**class** SavingAccount

{

**static** **double** AnnualIntrestRate=0.04, MontlyIntrest;

**private** **double** SavingBalance;

**double** getSB() {

**return** SavingBalance;

}

**void** setSB(**double** sB) {

**this**.SavingBalance = sB;

}

**void** calculateMontlyIntrest()

{

MontlyIntrest = (SavingBalance \* AnnualIntrestRate) / 12;

SavingBalance = SavingBalance + *MontlyIntrest*;

System.***out***.println(" MONTLY INTREST : "+MontlyIntrest);

System.***out***.println(" MODIFIED SAVING BALANCE : "+SavingBalance);

}

**static** **void** ModifieIntrest(**double** AI)

{

*AnnualIntrest*=AI;

System.***out***.println("\n MODIFIED ANNUAL INTREST : "+AnnualIntrestRate);

}

}

**public** **class** AccountTest{

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println(" FOR ANNUAL INTREST 0.04\n");

SavingAccount saver1=**new** SavingAccount();

SavingAccount saver2=**new** SavingAccount();

System.***out***.println(" ====FOR SAVER 1====");

saver1.setSB(2000);

System.***out***.println("\n CURRENT BALANCE OF SAVER 1 "+saver1.getSB());

saver1.calculateMontlyIntrest();

saver1.ModifieIntrest(0.05);

System.***out***.println(" MODIFIED INTREST OF SAVER 1");

saver1.calculateMontlyIntrest();

System.***out***.println("\n ====FOR SAVER 2====");

saver2.setSB(3000);

System.***out***.println("\n CURRENT BALANCE OF SAVER 2 "+saver2.getSB());

saver2.calculateMontlyIntrest();

saver2.ModifieIntrest(0.05);

System.***out***.println(" MODIFIED INTREST OF SAVER 2");

saver2. calculateMontlyIntrest();

}

}

**Output:**

FOR ANNUAL INTREST 0.04

====FOR SAVER 1====

CURRENT BALANCE OF SAVER 1 2000.0

MONTLY INTREST : 6.666666666666667

MODIFIED SAVING BALANCE : 2006.6666666666667

MODIFIED ANNUAL INTREST : 0.05

MODIFIED INTREST OF SAVER 1

MONTLY INTREST : 8.361111111111112

MODIFIED SAVING BALANCE : 2015.0277777777778

====FOR SAVER 2====

CURRENT BALANCE OF SAVER 2 3000.0

MONTLY INTREST : 12.5

MODIFIED SAVING BALANCE : 3012.5

MODIFIED ANNUAL INTREST : 0.05

MODIFIED INTREST OF SAVER 2

MONTLY INTREST : 12.552083333333334

MODIFIED SAVING BALANCE : 3025.052083333333

**Experiment No 3**

**Problem Statement:**

Create Vehicle Interface with name, maxPassanger, and maxSpeed variables. Create LandVehicle and SeaVehicleInteface from Vehicle interface. LandVehicle has numWheels variable and drive method. SeaVehicle has displacement variable and launch method. Create Car class from LandVehicle, HoverCraft from LandVehicle and SeaVehicle interface. Also create Ship from SeaVehicle. Provide additional methods in HoverCraft as enterLand and enterSea. Similarly provide other methods for class Car and Ship. Demonstrate all classes in a application.

**//Source Code**

//creating Interface Vehicle

interface Vehicle{

String name="Hero";

int MaxPessanger=5;

int MaxSpeed=120;

}

//Creating interface LandVehicle

interface LandVehicle extends Vehicle{

int nowheels=4;

public abstract void Drive();

}

//Creating interface SeaVehicle

interface SeaVehicle extends Vehicle{

int Displacement=200;

public abstract void Launch();

}

//Creating class Car which implements interface LAndVehicle

class Car implements LandVehicle

{

public void Drive(){

System.out.println("Drive Slowely And Be Safe");

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

}

public void Color(){

System.out.println("CAR color is Red");

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

}

public void Seats(){

System.out.println("There are 4 Seats");

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

}

}

//Creating class Ship which implements interface SaeVehicle

class Ship implements SeaVehicle

{

public void Launch(){

System.out.println("Launch your Vehivle in proper way");

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

}

public void Length() {

System.out.println("Length of Ship is: 50");

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

}

public void Rooms() {

System.out.println("Number of Rooms: 50");

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

}

}

//Creating class Craft which implements interface SaeVehicle and LandVehicle

class Craft implements LandVehicle,SeaVehicle

{

public void Drive(){

System.out.println("You are Driving on the Land as well as in Water");

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

}

public void Launch(){

System.out.println("Launch your Vehcle in proper Way");

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

}

public void OnLand() {

System.out.println("Are you Driving on Land");

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

}

public void InSea() {

System.out.println("Are you Driving in Sea");

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

}

}

public class VehicleInfo

{ public static void main(String args[])

{ Car c=new Car();

c.Drive();

c.Color();

c.Seats();

Ship s=new Ship();

s.Launch();

s.Length();

s.Rooms();

Craft ch=new Craft();

ch.Drive();

ch.Launch();

ch.OnLand();

ch.InSea();

}

}

**Output:**

C:\Users\Anis>cd C:\java programs\Interface

C:\java programs\Interface>javac VehicleInfo.java

C:\java programs\Interface>java VehicleInfo

Drive Slowely And Be Safe

----------------------

CAR color is Red

----------------------

There are 4 Seats

----------------------

Launch your Vehivle in proper way

----------------------

Length of Ship is: 50

----------------------

Number of Rooms: 50

----------------------

You are Driving on the Land as well as in Water

----------------------

Launch your Vehcle in proper Way

----------------------

Are you Driving on Land

----------------------

Are you Driving in Sea

----------------------

**Experiment No 4**

**Problem Statement:**

Create abstract class Shape which has instance variables side, area and perimeter

And methods calculateArea(), calculatePerimeter() as abstract methods and display() as concrete method. Write subclasses which extend Shape class like Triangle, Rectangle, Circle, Cube and Squere and override abstract methods and display methods in subclass take instance variable if needed as per the formula. And use parameterized constructor to initialize instance variables using “this” reference variable

Write Test class and Create a reference variable of Shape which will hold the objects of all the sub classes and calculate respective area, perimeter and display the results.

**//Source Code**

//Creating abstract class Shape

**public** **abstract** **class** Shape {

**public** **int** side;

**public** **double** area;

**public** **double** perimeter;

//abstract methods

**abstract** **public** **void** calculateArea();

**abstract** **public** **void** calculatePerimeter();

**public** **void** Display()

{

System.***out***.println("AREA : "+area);

System.***out***.println("PERIMETER : "+perimeter);

}

}

**public** **class** Triangle **extends** Shape {

**int** side1,side2,side3;

**public** Triangle(**int** s1,**int** s2,**int** s3)

{

side1=s1;

side2=s2;

side3=s3;

}

**public** **void** calculateArea()

{ **double** s;

s=(side1+side2+side3)\*0.5;

**double** t=s\*(s-side1)\*(s-side2)\*(s-side3);

area=Math.*sqrt*(t);

}

**public** **void** calculatePerimeter() {

perimeter=side1+side2+side3;

}

}

**public** **class** Rectangle **extends** Shape{

**int** length,weidth;

**public** Rectangle(**int** length,**int** weidth)

{

**this**.length=length;

**this**.weidth=weidth;

}

**public** **void** calculateArea() {

area=length\*weidth;

}

**public** **void** calculatePerimeter()

{

perimeter=2\*(length+weidth);

}

}

**public** **class** Cube **extends** Shape{

**int** side;

**public** Cube(**int** side) {

**this**.side=side;

}

**public** **void** calculateArea() {

area=6\*side\*side;

}

**public** **void** calculatePerimeter() {

perimeter=12\*side;

}

}

**public** **class** Square **extends** Shape {

**int** side;

**public** Square(**int** side)

{

**this**.side=side;

}

**public** **void** calculateArea() {

area=side\*side;

}

**public** **void** calculatePerimeter() {

perimeter=4\*side;

}

}

//Creating class Test

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Triangle t=**new** Triangle(15,15,20);

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

t.calculateArea();

t.calculatePerimeter();

t.Display();

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

Rectangle r=**new** Rectangle(15,20);

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

r.calculateArea();

r.calculatePerimeter();

r.Display();

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

Cube c=**new** Cube(20);

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

c.calculateArea();

c.calculatePerimeter();

c.Display();

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

Square s=**new** Square(20);

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

s.calculateArea();

s.calculatePerimeter();

s.Display();

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

}

}

**Output:**

--------TRIANGLE---------

AREA : 111.80339887498948

PERIMETER : 50.0

--------RECTANGLE---------

AREA : 300.0

PERIMETER : 70.0

--------CUBE---------

AREA : 2400.0

PERIMETER : 240.0

--------SQUARE---------

AREA : 400.0

PERIMETER : 80.0

**Experiment No 5**

**Problem Statement:**

Develop a mathematical package for Statistical operations like Mean, Median, Average, Standard deviation. Create a sub package in the math package -convert. In “convert” package provide classes to convert decimal to octal, binary, hex and vice-versa. Develop application program to use this package, and build executable jar file of it.

**//Source Code**

//Creating math Package

**package** math;

//class Mean of package math

**public** **class** Mean

{

**public** **double** mean=0;

**public** **void** mean(**int** arr[],**int** n)

{

**int** sum=0;

System.***out***.println("GIVER ARRAY : ");

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

{

System.***out***.println(arr[i]);

}

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

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

{

sum=sum+arr[i];

}

mean=(**double**)sum/(**double**)n;

System.***out***.println("MEAN OF ARRAY : "+mean);

}

}

**package** math;

//class Median of package math

**public** **class** Median

{

**double** m=0;

**public** **void** CalculateMedian(**int** arr[],**int** n)

{

**if**(n%2==1){

m=arr[(n+1)/2-1];

}

**else**{

m=(arr[n/2-1]+arr[n/2])/2;

}

System.***out***.println("Median Of Array :"+m);

}

}

//Create Subpackage of math (convert)

**package** math.convert;

// BinaryToDecimal class of convert package

**public** **class** BinaryToDecimal

{

**public** **void** BinToDec(**int** Binary)

{

**int** n=0,decimal=0,m;

m=decimal;

**while**(**true**)

{

**if**(Binary==0){

**break**;

}

**else**{

**int** temp=Binary%10;

decimal=(**int**) (decimal+temp\*Math.*pow*(2, n));

Binary=Binary/10;

n++;

}

}

System.***out***.println("Binary to Decimal Conversion is : "+decimal);

}

}

**package** math.convert;

//DecimalToHex class of convert package

**public** **class** DecimalToHex

{

**public** **void** dectohex(**int** decimal)

{

**int** rem,m;

String hex="";

**char** hexchar[]= {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

m=decimal;

**while**(decimal>0)

{

rem=decimal%16;

hex=hexchar[rem]+hex;

decimal=decimal/16;

}

System.***out***.println("Decimal to HexaDecimal Conversion is : "+hex);

}

}

//Creating a TestPack class which include package math and subpackage convert

**import** math.\*;

**import** math.Mean;

**import** math.convert.BinaryToDecimal;

**import** math.convert.DecimalToHex;

**public** **class** TestPack {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** arr[]= {20,23,45,2,1,33};

**int** n=6;

//Calculate Mean

Mean m=**new** Mean();

m.mean(arr,n);

//Calculate Median

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

Median md=**new** Median();

md.CalculateMedian(arr, n);

//Binary to Decimal conversion

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

BinaryToDecimal b=**new** BinaryToDecimal();

b.BinToDec(1010);

//Decimal to HExadecimla conversion

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

DecimalToHex h=**new** DecimalToHex();

h.dectohex(12);

}

}

**Output:**

GIVER ARRAY :

20

23

45

2

1

33

-----------------------------

MEAN OF ARRAY : 20.666666666666668

-----------------------------

Median Of Array :23.0

-----------------------------

Binary to Decimal Conversion is : 10

-----------------------------

Decimal to HexaDecimal Conversion is : C

**Experiment No 6**

**Problem Statement:**

Develop application which can handle any 5 combination of predefined compile time and runtime exceptions using multiple catch blocks. Use throws and finally keywords as well.

**//Source Code**

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** javax.security.auth.login.AccountException;

**public** **class** Exception\_Handling {

**void** CheckedExc()**throws** IOException

{

FileInputStream file=**null**;

file=**new** FileInputStream("E:/x.txt");

file.close();

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

}

**void** UnCheckedExc()

{

**try** {

CheckedExc();

**int** n1=20;

**int** n2=30;

**int** res=n1/n2;

System.***out***.println(res);

String s="XYZ";

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

**int** arr[]= {1,2,3,4};

System.***out***.println(arr[5]);

String s1=**null**;

System.***out***.println(s1.length());

}

**catch**(NumberFormatException e)

{

System.***out***.println("Number Format Exception Occurs ");

}

**catch**(IOException e)

{

System.***out***.println("Input Output Exception ");

}

**catch**(ArrayIndexOutOfBoundsException e)

{

System.***out***.println("Array Index Out Of Bounds Exception ");

}

**catch**(ArithmeticException e)

{

System.***out***.println("Arithmetic Exception ");

}

**catch**(NullPointerException e)

{

System.***out***.println("Null pointer Exception");

}

**finally**

{

System.***out***.println("Finally IO\_Exception Handeled");

}

}

**public** **static** **void** main(String[] args) {

Exception\_Handling e=**new** Exception\_Handling();

e.UnCheckedExc();

System.***out***.println("END OF PROGRAM");

}

}

**Output:**

Input Output Exception

Finally IO\_Exception Handeled

END OF PROGRAM

**Experiment No 07**

**Problem Statement:**

Develop a BankAccount class which should contain all methods of Bank i.e. balanceEnquery(), withdraw(), transfer() and deposit(). You should create at least two objects of BankAccount using array and do all operations mentioned above. Also generate user defined exception LowBalanceException, NegetiveNumberException and PasswordMismatchException whenever required. To transfer amount from one account to another use two BankAccount objects.

**//Source Code**

**import** java.util.Scanner;

**public** **class** BankAccount {

**public** **double** Balance=0;

//Constructor

**public** BankAccount(**double** b)

{

Balance=b;

}

//Method for withdraw

**public** **void** WithDraw(**double** amount)**throws** LowBalanceException,NegativeNumberException

{

**if**(Balance<amount)

{

**throw** **new** LowBalanceException("Your Balance is Low");

}

**if**(amount<0)

{

**throw** **new** NegativeNumberException("You are entering Invalid Amount");

}

Balance=Balance-amount;

}

//Method for Deposit

**public** **void** Deposit(**double** amount) **throws** NegativeNumberException

{

**if**(amount<0)

{

**throw** **new** NegativeNumberException("You are entering Invalid amount");

}

Balance=Balance+amount;

}

//Method for checking Balance in Account

**public** **double** BalanceEnquery()

{

**return** Balance;

}

//Method for password checking

**public** **void** PasswordCheck(**int** pin)**throws** PasswordIncorrectException

{

**if**(pin!=2409)

{

**throw** **new** PasswordIncorrectException("You are entering Invalid pin !\n Please try again ");

}

}

//Method for Transfer Amount

**public** **void** Transfer(**double** amount)**throws** LowBalanceException,NegativeNumberException

{

**if**(Balance<amount)

{

**throw** **new** LowBalanceException("Your Balance is Low");

}

**if**(amount<0)

{

**throw** **new** NegativeNumberException("Ypu are entering Invalid Amount");

}

}

**public** **static** **void** main(String[] args)

{

Scanner sc=**new** Scanner(System.***in***);

BankAccount b[]=**new** BankAccount[3];

b[0]=**new** BankAccount(2000);

b[1]=**new** BankAccount(4000);

**int** password=2409;

**int** c=0;

**while**(c<=3)

{

System.***out***.println("\t WELCOME TO FEDRAL BANK");

System.***out***.println("1.BankAccount No. 1 \n2.BankAccount No. 2 \n3.Exit \n\nEnter Your Choice");

**int** i=sc.nextInt();

**if**(i>3||i<1)

{

**throw** **new** ArithmeticException("Wrong Choice Please try Again!");

}

c=i;

**int** j=0;

**while**(j!=6)

{

System.***out***.println("\t You are in BankAccount No."+(i));

System.***out***.println("1.Withdraw \n2.Deposit \n3.BalanceEnquery \n4.Transfer \n5.Exit \nEnter Your Choice");

**int** k=sc.nextInt();

j=k;

**switch**(k)

{

**case** 1:

**try** {

System.***out***.println("Enter Pin");

b[i].PasswordCheck(sc.nextInt());

System.***out***.println("Enter Amount to Withdraw");

**double** cash=sc.nextDouble();

b[i].WithDraw(cash);

}

**catch**(LowBalanceException s)

{

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

}

**catch**(NegativeNumberException s)

{

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

}

**catch**(PasswordIncorrectException s)

{

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

}

**break**;

**case** 2:

**try** {

System.***out***.println("Enter Pin ");

b[i].PasswordCheck(sc.nextInt());

System.***out***.println("Enter amount to Deposit");

**double** cash1=sc.nextDouble();

b[i].Deposit(cash1);

}

**catch**(NegativeNumberException s)

{

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

}

**catch**(PasswordIncorrectException s)

{

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

}

**break**;

**case** 3:

**try** {

System.***out***.println("Enter Pin ");

b[i].PasswordCheck(sc.nextInt());

}

**catch**(PasswordIncorrectException s)

{

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

}

System.***out***.println("Your Current Balance :"+b[i].BalanceEnquery());

**break**;

**case** 4:

**try** {

System.***out***.println("Enter Pin ");

b[i].PasswordCheck(sc.nextInt());

System.***out***.println("Please enter Amount to Transfer");

**double** t=sc.nextDouble();

b[i].Transfer(t);

**if**(i==0)

{

b[i]=**new** BankAccount(b[i].BalanceEnquery()-t);

b[i+1]=**new** BankAccount(b[i+1].BalanceEnquery()+t);

}

**else** **if**(i==1)

{

b[i]=**new** BankAccount(b[i].BalanceEnquery()-t);

b[i-1]=**new** BankAccount(b[i-1].BalanceEnquery()+t);

}

System.***out***.println(" Cash Transfer Sussessfully");

}

**catch**(LowBalanceException s)

{

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

}

**catch**(NegativeNumberException s)

{

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

}

**catch**(PasswordIncorrectException s)

{

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

}

**break**;

**case** 5:

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

**break**;

**default**:

System.***out***.println("Wrong Choice!");

**break**;

}

j++;

}

c++;

}

}

}

**Output:**

WELCOME TO FEDRAL BANK

1.BankAccount No. 1

2.BankAccount No. 2

3.Exit

Enter Your Choice

1

You are in BankAccount No.1

1.Withdraw

2.Deposit

3.BalanceEnquery

4.Transfer

5.Exit

Enter Your Choice

1

Enter Pin

2409

Enter Amount to Withdraw

20000

LowBalanceException: Your Balance is Low

You are in BankAccount No.1

1.Withdraw

2.Deposit

3.BalanceEnquery

4.Transfer

5.Exit

Enter Your Choice

3

Enter Pin

2221

PasswordIncorrectException: You are entering Invalid pin !

Please try again

Your Current Balance :4000.0

You are in BankAccount No.1

1.Withdraw

2.Deposit

3.BalanceEnquery

4.Transfer

5.Exit

Enter Your Choice

5

Exit

WELCOME TO FEDRAL BANK

1.BankAccount No. 1

2.BankAccount No. 2

3.Exit

**Experiment No 08**

**Problem Statement:**

Use Two Character file Stream classes to copy & write Characters to a File named “input.txt” into a file “Output.txt”.

**//Source Code**

import java.io.\*;

public class CopyChar

{

public static void main(String[] args) {

File in = new File("input.txt");

File out = new File("Output.txt");

FileReader fr=null;

FileWriter fw=null;

try {

fr = new FileReader(in);

fw = new FileWriter(out);

// Read & write till end of File

int ch;

while ((ch = fr.read()) != -1)

{ fw.write(ch);

}

}catch (IOException e)

{ System.out.println(e);

System.out.println(-1);

}

finally {

try {

fr.close();

fw.close();

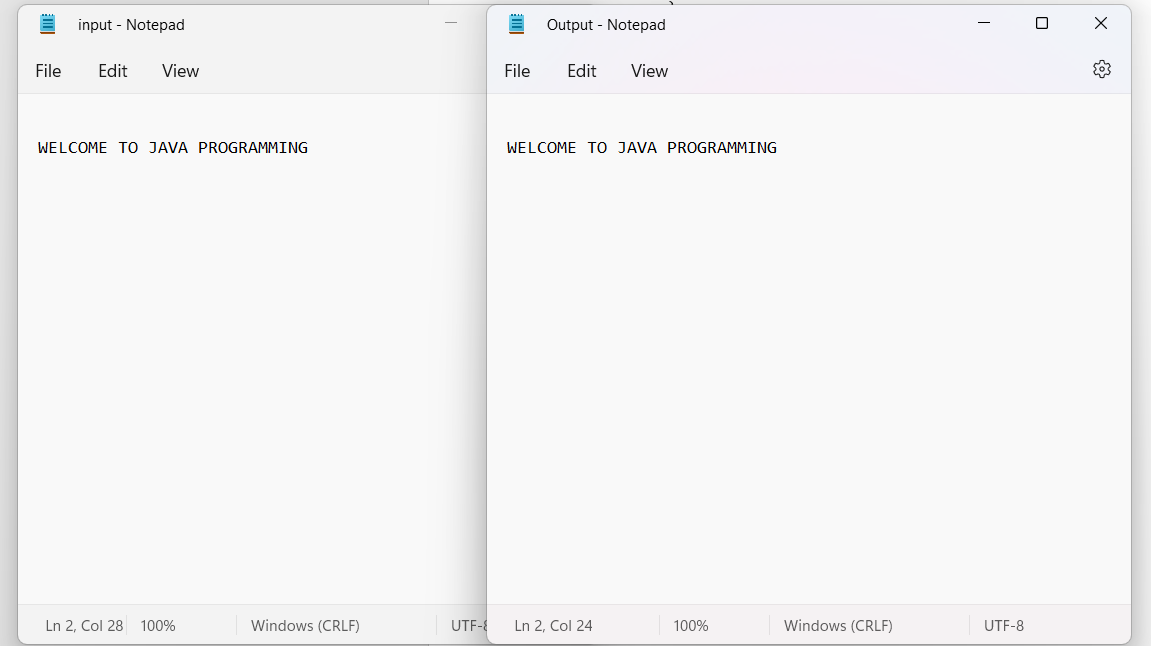
} catch (IOException e) { }

}

}

}

**Output:**



**Experiment No 09**

**Problem Statement:**

Use Two file Byte Stream classes to copy & write Bytes to a File named “in.txt” into a file “out.txt”.

**//Source Code**

**import** java.io.\*;

**public** **class** CopyByte {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

FileInputStream in=**null**;

FileOutputStream out=**null**;

**byte** b;

**try** {

in=**new** FileInputStream("in.txt");

out=**new** FileOutputStream("out.txt");

**do** {

b=(**byte**)in.read();

out.write(b);

}

**while**(b!=-1);

}

**catch**(IOException e){

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

}

**finally**

{

**try** {

in.close();

out.close();

}

**catch**(IOException e)

{

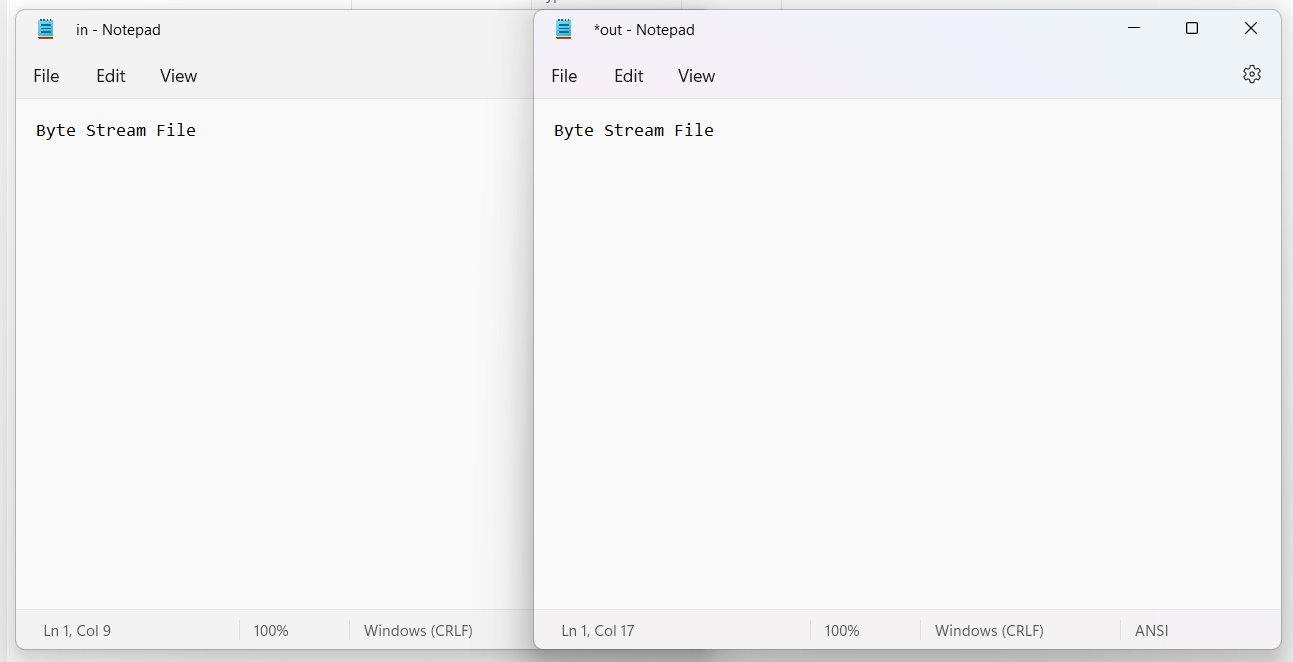
}

}

}

}

**Output:**



**Experiment No 10**

**Problem Statement:**

Develop a Swing GUI based standard calculator program. Use event handling, Layout of swing package.

**//Source Code**

import java.awt.Color;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.lang.module.ModuleDescriptor.Builder;

import javax.swing.ButtonGroup;

import javax.swing.ButtonModel;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JRadioButton;

import javax.swing.JTextField;

import javax.swing.SwingConstants;

import javax.swing.text.AttributeSet.ColorAttribute;

import java.awt.event.ActionListener;

public class Calculator implements ActionListener

{ // Creating Variables for our Calculations

double Number,Answer;

int Calculation;

JFrame frame;

JLabel label=new JLabel();

JTextField textField=new JTextField();

JRadioButton onRadioButton=new JRadioButton("On");

JRadioButton offRadioButton=new JRadioButton("Off");

JButton buttonZero=new JButton("0");

JButton buttonOne=new JButton("1");

JButton buttonTwo=new JButton("2");

JButton buttonThree=new JButton("3");

JButton buttonFour=new JButton("4");

JButton buttonFive=new JButton("5");

JButton buttonSix=new JButton("6");

JButton buttonSeven=new JButton("7");

JButton buttonEight=new JButton("8");

JButton buttonNine=new JButton("9");

JButton buttonDot=new JButton(".");

JButton buttonClear=new JButton("C");

JButton buttonDelete=new JButton("<-");

JButton buttonEqual=new JButton("=");

JButton buttonMul=new JButton("x");

JButton buttonDiv=new JButton("/");

JButton buttonPlus=new JButton("+");

JButton buttonMinus=new JButton("-");

JButton buttonSquare=new JButton("x\u00B2");

JButton buttonReciprocal=new JButton("1/x");

JButton buttonSqrt=new JButton("\u221A");

Calculator()

{ PrepareGUI();

AddComponent();

AddActionEvent();

}

public static void main(String[] args)

{ new Calculator();

}

public void PrepareGUI()

{ frame=new JFrame();

frame.setTitle("Calculator");

frame.setSize(300,490);

frame.getContentPane().setLayout(null);

frame.getContentPane().setBackground(Color.black);

frame.setResizable(false);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void AddComponent(){

label.setBounds(250, 0, 50, 50);

label.setForeground(Color.white);

frame.add(label);

textField.setBounds(10, 40, 270, 40);

textField.setFont(new Font("Arial",Font.BOLD,20));

textField.setEditable(false);

textField.setHorizontalAlignment(SwingConstants.RIGHT);

frame.add(textField);

onRadioButton.setBounds(10, 95, 60, 40);

onRadioButton.setSelected(true);

onRadioButton.setFont(new Font("Arial",Font.BOLD,14));

onRadioButton.setBackground(Color.black);

onRadioButton.setForeground(Color.white);

frame.add(onRadioButton);

offRadioButton.setBounds(10, 120, 60, 40);

offRadioButton.setSelected(false);

offRadioButton.setFont(new Font("Arial",Font.BOLD,14));

offRadioButton.setBackground(Color.black);

offRadioButton.setForeground(Color.white);

frame.add(offRadioButton);

ButtonGroup buttonGroup=new ButtonGroup();

buttonGroup.add(onRadioButton);

buttonGroup.add(offRadioButton);

buttonSeven.setBounds(10, 230, 60, 40);

buttonSeven.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonSeven);

buttonEight.setBounds(80, 230, 60, 40);

buttonEight.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonEight);

buttonNine.setBounds(150, 230, 60, 40);

buttonNine.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonNine);

buttonFour.setBounds(10, 290, 60, 40);

buttonFour.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonFour);

buttonFive.setBounds(80, 290, 60, 40);

buttonFive.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonFive);

buttonSix.setBounds(150, 290, 60, 40);

buttonSix.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonSix);

buttonOne.setBounds(10, 350, 60, 40);

buttonOne.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonOne);

buttonTwo.setBounds(80, 350, 60, 40);

buttonTwo.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonTwo);

buttonThree.setBounds(150, 350, 60, 40);

buttonThree.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonThree);

buttonDot.setBounds(150, 410, 60, 40);

buttonDot.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonDot);

buttonZero.setBounds(10, 410, 60, 40);

buttonZero.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonZero);

buttonEqual.setBounds(220, 350, 60, 100);

buttonEqual.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonEqual);

buttonDiv.setBounds(220, 110, 60, 40);

buttonDiv.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonDiv);

buttonSqrt.setBounds(10, 170, 60, 40);

buttonSqrt.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonSqrt);

buttonMul.setBounds(220, 230, 60, 40);

buttonMul.setFont(new Font("Arial",Font.BOLD,20));

buttonMul.setBackground(new Color(239,188,2));

frame.add(buttonMul);

buttonMinus.setBounds(220, 170, 60, 40);

buttonMinus.setFont(new Font("Arial",Font.BOLD,20));

buttonMinus.setBackground(new Color(239,188,2));

frame.add(buttonMinus);

buttonPlus.setBounds(220, 290, 60, 40);

buttonPlus.setFont(new Font("Arial",Font.BOLD,20));

buttonPlus.setBackground(new Color(239,188,2));

frame.add(buttonPlus);

buttonSquare.setBounds(80, 170, 60, 40);

buttonSquare.setFont(new Font("Arial",Font.BOLD,20));

frame.add(buttonSquare);

buttonReciprocal.setBounds(150, 170, 60, 40);

buttonReciprocal.setFont(new Font("Arial",Font.BOLD,15));

frame.add(buttonReciprocal);

buttonDelete.setBounds(150, 110, 60, 40);

buttonDelete.setFont(new Font("Arial",Font.BOLD,12));

buttonDelete.setBackground(Color.red);

buttonDelete.setForeground(Color.white);

frame.add(buttonDelete);

buttonClear.setBounds(80, 110, 60, 40);

buttonClear.setFont(new Font("Arial",Font.BOLD,12));

buttonClear.setBackground(Color.red);

buttonClear.setForeground(Color.white);

frame.add(buttonClear);

}

public void AddActionEvent()

{ // Registering ActionListner to buttons

onRadioButton.addActionListener(this);

offRadioButton.addActionListener(this);

buttonClear.addActionListener(this);

buttonDelete.addActionListener(this);

buttonDiv.addActionListener(this);

buttonSqrt.addActionListener(this);

buttonSquare.addActionListener(this);

buttonReciprocal.addActionListener(this);

buttonMinus.addActionListener(this);

buttonSeven.addActionListener(this);

buttonEight.addActionListener(this);

buttonNine.addActionListener(this);

buttonMul.addActionListener(this);

buttonFour.addActionListener(this);

buttonFive.addActionListener(this);

buttonSix.addActionListener(this);

buttonPlus.addActionListener(this);

buttonOne.addActionListener(this);

buttonTwo.addActionListener(this);

buttonThree.addActionListener(this);

buttonEqual.addActionListener(this);

buttonZero.addActionListener(this);

buttonDot.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{ Object source=e.getSource();

if(source==onRadioButton){

enable(); // Calling enable() Method

}

else if(source==offRadioButton)

{ disable(); // Calling disable() Method

}

else if(source==buttonClear)

{ label.setText("");

textField.setText("");

}else if(source==buttonDelete)

{ int length=textField.getText().length();

int number=length-1;

if(length>0){

StringBuilder back=new StringBuilder(textField.getText());

back.deleteCharAt(number);

textField.setText(back.toString());

}

if(textField.getText().endsWith("")){

label.setText("");

} }

else if(source==buttonZero){

if(textField.getText().equals("0")){

return;

}

else{

textField.setText(textField.getText()+"0");

}

}

else if(source==buttonOne){

textField.setText(textField.getText()+"1");

}

else if(source==buttonTwo){

textField.setText(textField.getText()+"2");

}

else if(source==buttonThree){

textField.setText(textField.getText()+"3");

}

else if(source==buttonFour){

textField.setText(textField.getText()+"4");

}

else if(source==buttonFive){

textField.setText(textField.getText()+"5");

}

else if(source==buttonSix){

textField.setText(textField.getText()+"6");

}

else if(source==buttonSeven){

textField.setText(textField.getText()+"7");

}

else if(source==buttonEight){

textField.setText(textField.getText()+"8");

}

else if(source==buttonNine){

textField.setText(textField.getText()+"9");

}

else if(source==buttonDot){

if(textField.getText().contains("."))

return;

else

textField.setText(textField.getText()+".");

}

else if(source==buttonPlus){

String str=textField.getText();

Number=Double.parseDouble(textField.getText());

textField.setText("");

label.setText(str+"+");

Calculation=1;

}

else if(source==buttonMinus){

String str=textField.getText();

Number=Double.parseDouble(textField.getText());

textField.setText("");

label.setText(str+"-");

Calculation=2;

}

else if(source==buttonMul){

String str=textField.getText();

Number=Double.parseDouble(textField.getText());

textField.setText("");

label.setText(str+"X");

Calculation=3;

}

else if(source==buttonDiv){

String str=textField.getText();

Number=Double.parseDouble(textField.getText());

textField.setText("");

label.setText(str+"/");

Calculation=4;

}

else if(source==buttonSqrt){

Number=Double.parseDouble(textField.getText());

Double sqrt=Math.sqrt(Number);

textField.setText(Double.toString(sqrt));

}

else if(source==buttonSquare){

String str=textField.getText();

Number=Double.parseDouble(textField.getText());

Double Square=Math.pow(Number, 2);

String string=Double.toString(Square);

if (string.endsWith(".0")) {

textField.setText(string.replace(".0", ""));

}

else{

textField.setText(string);

}

label.setText("(sqr)"+str);

}

else if(source==buttonReciprocal){

Number=Double.parseDouble(textField.getText());

double reciprocal=1/Number;

String string=Double.toString(reciprocal);

if(string.endsWith(".0")){

textField.setText(string.replace(".0", ""));

}

else{

textField.setText(string);

}

}

else if(source==buttonEqual){

// Setting Functionality for Equal (=) button

switch(Calculation){

case 1:

Answer=Number+Double.parseDouble(textField.getText());

if(Double.toString(Answer).endsWith(".0")){

textField.setText(Double.toString(Answer).replace(".0", ""));

}

else{

textField.setText(Double.toString(Answer));

}

label.setText("");

break;

case 2:

Answer=Number-Double.parseDouble(textField.getText());

if(Double.toString(Answer).endsWith(".0")){

textField.setText(Double.toString(Answer).replace(".0", ""));

}else{

textField.setText(Double.toString(Answer));

}

label.setText("");

break;

case 3:

Answer=Number\*Double.parseDouble(textField.getText());

if(Double.toString(Answer).endsWith(".0")){

textField.setText(Double.toString(Answer).replace(".0", ""));

}

else{

textField.setText(Double.toString(Answer));

}

label.setText("");

break;

case 4:

Answer=Number/Double.parseDouble(textField.getText());

if(Double.toString(Answer).endsWith(".0")){

textField.setText(Double.toString(Answer).replace(".0", ""));

}

else{

textField.setText(Double.toString(Answer));

}

label.setText("");

break;

}

}

}

public void enable()

{

onRadioButton.setEnabled(false);

offRadioButton.setEnabled(true);

textField.setEnabled(true);

label.setEnabled(true);

buttonClear.setEnabled(true);

buttonDelete.setEnabled(true);

buttonDiv.setEnabled(true);

buttonSqrt.setEnabled(true);

buttonSquare.setEnabled(true);

buttonReciprocal.setEnabled(true);

buttonMinus.setEnabled(true);

buttonSeven.setEnabled(true);

buttonEight.setEnabled(true);

buttonNine.setEnabled(true);

buttonMul.setEnabled(true);

buttonFour.setEnabled(true);

buttonFive.setEnabled(true);

buttonSix.setEnabled(true);

buttonPlus.setEnabled(true);

buttonOne.setEnabled(true);

buttonTwo.setEnabled(true);

buttonThree.setEnabled(true);

buttonEqual.setEnabled(true);

buttonZero.setEnabled(true);

buttonDot.setEnabled(true);

}

public void disable()

{ onRadioButton.setEnabled(true);

offRadioButton.setEnabled(false);

textField.setText("");

label.setText("");

buttonClear.setEnabled(false);

buttonDelete.setEnabled(false);

buttonDiv.setEnabled(false);

buttonSqrt.setEnabled(false);

buttonSquare.setEnabled(false);

buttonReciprocal.setEnabled(false);

buttonMinus.setEnabled(false);

buttonSeven.setEnabled(false);

buttonEight.setEnabled(false);

buttonNine.setEnabled(false);

buttonMul.setEnabled(false);

buttonFour.setEnabled(false);

buttonFive.setEnabled(false);

buttonSix.setEnabled(false);

buttonPlus.setEnabled(false);

buttonOne.setEnabled(false);

buttonTwo.setEnabled(false);

buttonThree.setEnabled(false);

buttonEqual.setEnabled(false);

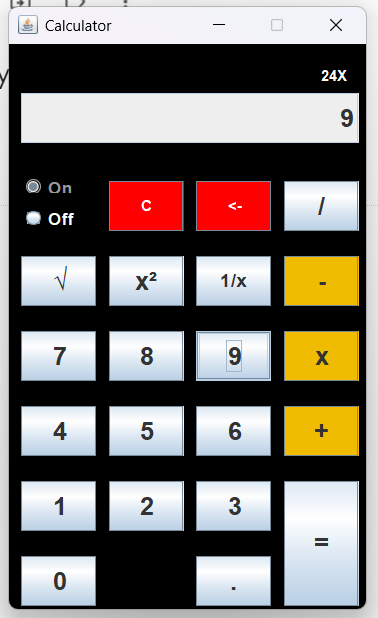
buttonZero.setEnabled(false);

buttonDot.setEnabled(false);

}

}

**Output:**



**Experiment No 11**

**Problem Statement:**

Create Stop Watch with Swing GUI and Multithreading. Provide Facility for Lap Counting.

**//Source Code**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.text.DecimalFormat;

import java.text.NumberFormat;

public class StopWatch extends JFrame implements ActionListener,Runnable

{

JLabel disp,lap1,lap2,lap3;

JButton btn,lapbtn;

boolean stop=false;

int i,j,k,l;

static int count=0;

public StopWatch()

{ disp=new JLabel();

disp.setBounds(70,15,110,15);

btn=new JButton("start");

btn.setBounds(250,20,80,20);

disp.setFont(new Font("Helvetica",Font.PLAIN,20));

disp.setForeground(Color.red);

disp.setBackground(Color.cyan);

Container c=getContentPane();

c.setLayout(null);

c.add(disp);

c.add(btn);

btn.addActionListener(this);

lapbtn=new JButton("Lap");

lapbtn.setBounds(180,60,70,15);

c.add(lapbtn);

lap1=new JLabel("");

lap1.setBounds(80,110,80,25);

c.add(lap1);

lap2=new JLabel("");

lap2.setBounds(80,150,80,25);

c.add(lap2);

lap3=new JLabel("");

lap3.setBounds(80,180,80,25);

c.add(lap3);

lapbtn.addActionListener(this);

}

public void run()

{

for(i=0;;i++)

{ for(j=0;j<60;j++)

{ for(k=0;k<60;k++)

{ for(l=0;l<100;l++)

{ if(stop)

{

break;

}

NumberFormat nf=new DecimalFormat("00");

disp.setText(nf.format(i)+":"+nf.format(j)+":"+nf.format(k)+":"+nf.format(l));

try

{

Thread.sleep(10);

}

catch(Exception e)

{

}

}

}

}

}

}

public void actionPerformed(ActionEvent ae)

{

Thread t=new Thread(this);

if(ae.getActionCommand().equals("start"))

{ t.start();

btn.setText("stop");

}

if(ae.getActionCommand().equals("stop"))

{ stop=true;

}

if(ae.getActionCommand().equals("Lap"))

{

String label=null;

if(count==0)

{

label=disp.getText();

lap1.setText(label);

}

if(count==1)

{

label=disp.getText();

lap2.setText(label);

}

if(count==2)

{

label=disp.getText();

lap3.setText(label);

}

count++;

}

}

public static void main(String args[])

{

StopWatch s=new StopWatch();

s.setSize(500,500);

s.setVisible(true);

s.setTitle("STOP WATCH");

s.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

**Output:** 