import java.util.\*;

class Constructures

{

public static void main(String args[])

{

menu();

}

static void menu()

{

Scanner s=new Scanner(System.in);

while(true)

{

System.out.println("Hello,welcome to Java Programming");

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

System.out.println("Enter any option you want from the following");

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

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

System.out.println("3.jumping");

System.out.println("4.exit");

int choice=s.nextInt();

switch(choice)

{

case 1:

while(true)

{

System.out.println("conditional");

System.out.println("select any one option");

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

System.out.println("2.if-else");

System.out.println("3.nested");

System.out.println("4.go back");

int select=s.nextInt();

switch(select)

{

case 1:

System.out.println("enter your age");

int age=s.nextInt();

if(age>=18)

{

System.out.println("You are eligible to vote");

}

break;

case 2:

System.out.println("nter your age");

int num=s.nextInt();

if(num%2==0)

{

System.out.println("you entered even number");

}

else

{

System.out.println("you entered odd number");

}

break;

case 3:

System.out.println("enter 1st number");

int a=s.nextInt();

System.out.println("enter 2nd number");

int b=s.nextInt();

System.out.println("enter 3rd number");

int c=s.nextInt();

if(a>b)

{

if(a<c)

{

System.out.println("1st one is Bigger");

}

else

{

System.out.println("3rd one is bigger");

}

}

else

{

if(b>c)

{

System.out.println("2nd one is bigger");

}

else

{

System.out.println("3rd one is bigger");

}

}

break;

case 4:

default:

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

}

break;

}

break;

case 2:

while(true)

{

System.out.println("looping");

System.out.println("select any one option");

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

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

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

System.out.println("4.for each");

System.out.println("5.go back");

int select=s.nextInt();

switch(select)

{

case 1:

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

int w=1;

while(w<=10)

{

System.out.println(w);

w++;

}

break;

case 2:

System.out.println("enter a number to find sum of squares of n");

int n=s.nextInint sum=0,r=1;

do

{

sum=sum+(r\*r);

r++;

}

while(r<=n);

System.out.println("sum of squares is="+sum);

break;

case 3:

System.out.println("enter a value to do factorial");

int f=s.nextInt();

int fact=1;

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

{

fact=fact\*i;

}

System.out.println("factorial of "+f+" !="+fact);

break;

case 4:

System.out.println("foreach");

int[] useArray={10,20,30,40,50};

System.out.println("printing array elements");

for(int num:useArray)

{

System.out.println(num);

}

break;

case 5:

break;

default:

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

}

break;

}

break;

case 3:

while(true)

{

System.out.println("jumping");

System.out.println("select any one option");

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

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

System.out.println("3.return");

System.out.println("4.go back");

int select=s.nextInt();

switch(select)

{

case 1:

System.out.println("break");

int n=10;

int b=5;

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

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

{

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

if(i==b)

{

break;

}

}

case 2:

System.out.println("CONTINUE");

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

for(int i=1;i<=10;i++)

{

if(i==5)

{

continue;

}

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

}

case 3:

System.out.println("executing return statement");

System.out.println("exit the program");

return;

case 4:

break;

default:

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

}

break;

}

break;

case 4:

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

System.exit(0);

s.close();

break;

default:

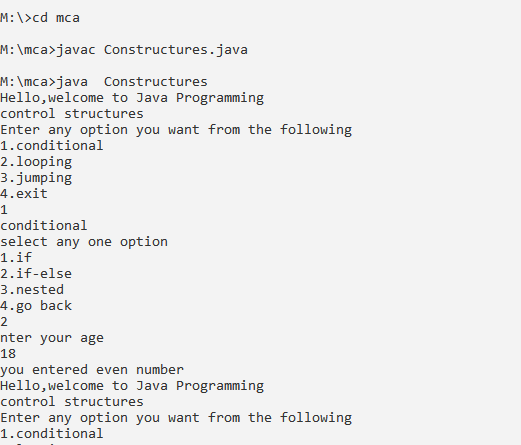
System.out.println("invalid option.please enter valid option”)

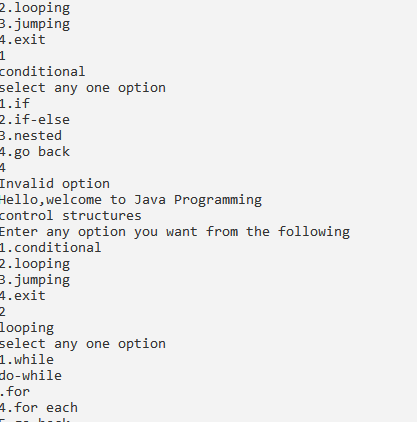
}

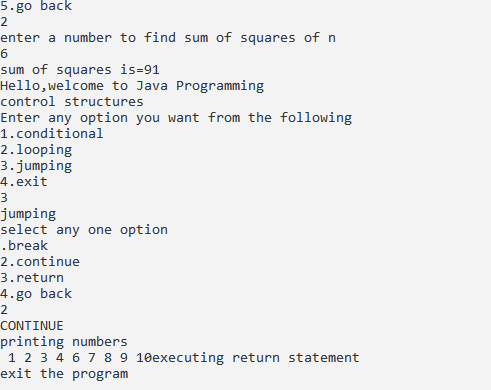
}

}

}







import java.util.Scanner;

public class Oops

{

public static void main(String args[])

{

menu();

}

static void menu()

{

Scanner scn=new Scanner(System.in);

while(true)

{

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

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

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

System.out.println("3.polymorphism");

System.out.println("4.inheritance");

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

System.out.println("choose your choice");

int choice=scn.nextInt();

switch(choice)

{

case 1:

B b=new B();

b.m1();

b.m2();

b.m3();

break;

case 2:

Scanner sc=new Scanner(System.in);

Factorial ft=new Factorial();

System.out.println("enter a number");

int n=sc.nextInt();

ft.setNum(n);

System.out.println(ft.getNum()+"!="+ft.fact());

break;

case 3:

Scanner s=new Scanner(System.in);

Add a=new Add();

System.out.println("enter first number");

int x=s.nextInt();

System.out.println("enter second number");

int y=s.nextInt();

a.sum(x,y);

System.out.println("enter first number");

int k=s.nextInt();

System.out.println("enter second number");

int l=s.nextInt();

System.out.println("enter third number");

int m=s.nextInt();

a.sum(k,l,m);

break;

case 4:

D d=new D();

d.m2();

break;

case 5:

System.out.println("existing the program bye ...!");

System.exit(0);

default:

System.out.println("please choose valid choice ");

break;

}

}

}

}

class B extends A

{

@Override

public void m1()

{

System.out.println("abstract method m1() defined in class B");

}

}

abstract class A

{

abstract void m1();

public void m2()

{

System.out.println("m2() defined in class A");

}

public void m3()

{

System.out.println("m3() defined in class A");

}

}

class Factorial

{

private int num;

public int getNum()

{

return num;

}

public void setNum(int num)

{

this.num=num;

}

public int fact()

{

int fact=1;

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

{

fact=fact\*i;

}

return fact;

}

}

class Add

{

public void sum(int a,int b)

{

int c=a+b;

System.out.println("sum of "+a+" and "+b+" "+c+" is :"+c);

}

public void sum(int a,int b,int c)

{

int sum=a+b+c;

System.out.println("sum of "+a+","+b+", "+c+" is:"+sum);

}

}

class C

{

public void m1()

{

System.out.println("m1() defined in class C");

}

}

class D extends C

{

public void m2()

{

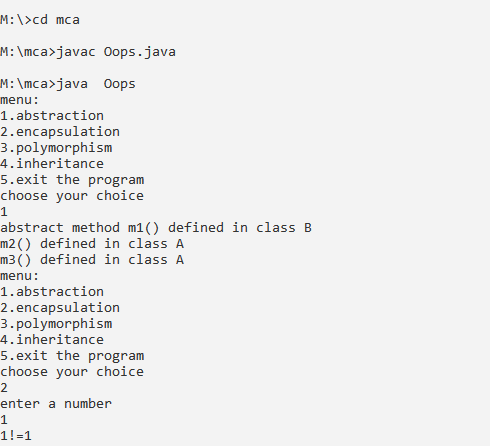
m1();

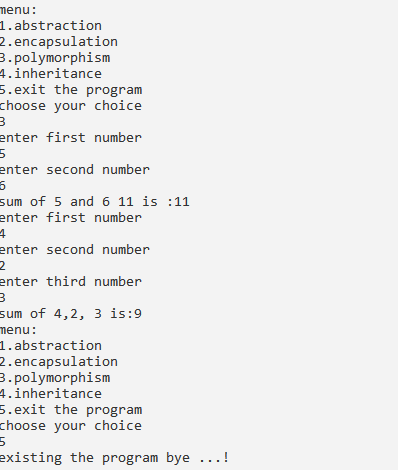
System.out.println("m2() defined in class D");

}

}

OUTPUT:





import java.util.\*;

class Methodoverloading

{

public static void main(String args [])

{

Scanner scn=new Scanner(System.in);

Multiplication mu=new Multiplication();

while(true)

{

System.out.println("1.multiplication of two numbers ");

System.out.println("2.multiplication of three numbers ");

System.out.println("3.exit ");

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

int choice=scn.nextInt();

switch(choice)

{

case 1:

System.out.println("enter first number ");

int n1=scn.nextInt();

System.out.println("enter second number ");

int n2=scn.nextInt();

mu.mul(n1,n2);

break;

case 2:

System.out.println("enter first number ");

int num1=scn.nextInt();

System.out.println("enter second number ");

int num2=scn.nextInt();

System.out.println("enter third number ");

int num3=scn.nextInt();

mu.mul(num1,num2,num3);

break;

case 3:

System.out.println("exit from the program ");

System.exit(0);

break;

default:

System.out.println(" enter invalid option ");

break;

}

}

}

}

class Multiplication

{

public void mul(int n1,int n2)

{

int result=n1\*n2;

System.out.println("product of"+n1+"and"+n2+"is:"+result);

}

public void mul(int num1, int num2, int num3)

{

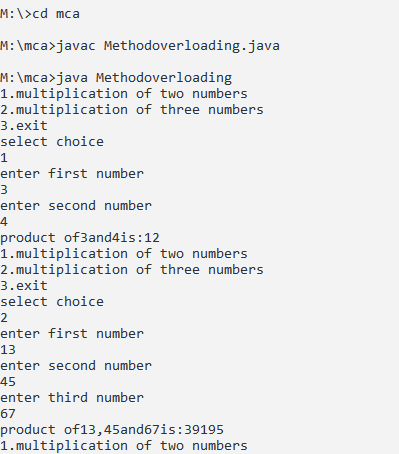
int mul=num1\*num2\*num3;

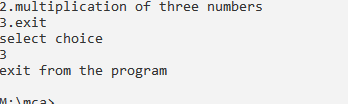
System.out.println("product of"+num1+","+num2+"and"+num3+"is:"+mul);

}

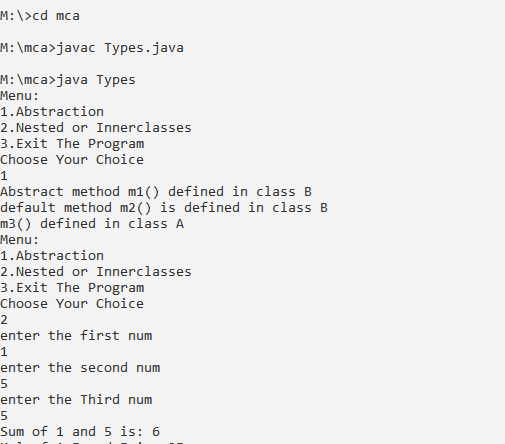
}

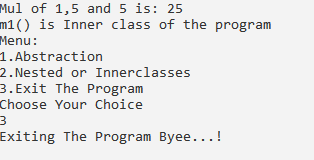
OUTPUT:





OUTPUT:





import java.util.Scanner;

public class InheriDemo {

public static void main(String[] args) {

menu();

}

static void menu() {

Scanner scn = new Scanner(System.in);

while(true) {

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

System.out.println("1.single inheritance");

System.out.println("2.multilevel inheritance");

System.out.println("3.multiple(interface)");

System.out.println("4.Hierarchical inheritance");

System.out.println("5.hybrid inheritance");

System.out.println("6.Exit the menu");

System.out.println("Choose Your Choice");

int choice = scn.nextInt();

switch(choice) {

case 1:

B b = new B();

b.m1();

b.m2();

b.m3();

b.m4();

break;

case 2:

Three t=new Three();

t.volume();

t.display();

Two t1=new Two();

t1.display();

t1.print();

break;

case 3:

Scanner s1=new Scanner(System.in);

System.out.println("enter the first num");

int n1=s1.nextInt();

System.out.println(" enter the Second num");

int n2=s1.nextInt();

System.out.println("enter the third num");

int n3=s1.nextInt();

Sub w=new Sub();

w.sum(n1,n2);

w.mult(n1,n2,n3);

w.divi1(n1,n2);

break;

case 4:

D d = new D();

d.m2();

E e=new E();

e.m3();

break;

case 5:

Earth e1=new Earth();

e1.double1();

Mars r1=new Mars();

r1.triple();

break;

case 6:

System.out.println("Exiting The Program Byee...!");

System.exit(0);

scn.close();

break;

default:

System.out.println("Please Choose Valid Choice");

break;

}

}

}

}

// single inheritance

class A {

public void m1() {

System.out.println("m1() defined in class A");

}

public void m2() {

System.out.println("m2() defined in class A");

}

}

class B extends A {

public void m3() {

System.out.println("m3() defined in class B");

}

public void m4() {

System.out.println("m4() is defined in class B");

}

}

// multilevel inheritance

class One {

public void print() {

System.out.println("I am the parent class");

}

}

class Two extends One {

public void display() {

System.out.println("I am the sub class of the one class");

}}

class Three extends Two {

public void volume() {

System.out.println("I am the sub class of the Two class");

}

}

// multiple inheritance

interface Sum {

public void sum(int a,int b);

}

interface Mul {

public void mult(int a,int b,int c);

}

interface Divi extends Sum,Mul {

public void divi1(int a,int b);

}

class Sub implements Divi {

public void sum(int a,int b) {

int c = a+b;

System.out.println("Sum of "+a+" and "+b+" is: "+c);

}

public void mult(int a,int b,int c) {

int mul=a\*b\*c;

System.out.println("mul"+a+","+b+"and"+c+"is:"+mul);

}

public void divi1(int a,int b) {

int div=a%b;

System.out.println("divi"+a+","+b+" is:"+div);

}

}

// hierarchical inheritance

class C {

public void m1() {

System.out.println("m1() defined in class c");

}

}

class D extends C {

public void m2() {

m1();

System.out.println("m2() defined in class D");

}

}

class E extends C {

public void m3() {

m1();

System.out.println("m3() defines in class E");

}

}

// hybrid inheritance

class SolarSystem {

public void single() {

System.out.println("I am the Solar class method");

}

}

class Earth extends SolarSystem {

public void double1() {

System.out.println("I am the Earth class method");

}

}

class Mars extends SolarSystem {

public void triple() {

System.out.println("I am the Mars class method");

}

}

class Moon extends Earth {

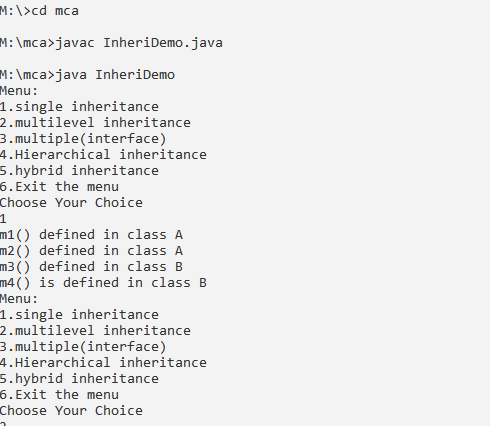
public void forth() {

System.out.println("I am the Moon class method");

}

}

OUTPUT:



import java.util.Scanner;

public class superDemo

{

public static void main(String args[])

{

menu();

}

static void menu()

{

Scanner scn=new Scanner(System.in);

while(true)

{

System.out.println("1.super keyword");

System.out.println("2.this keyword");

System.out.println("3.final keyword");

System.out.println("4.exit the program");

System.out.println("choose your choice");

int choice=scn.nextInt();

switch(choice)

{

case 1:

B b=new B();

b.display();

b.show();

break;

case 2:

Scanner s=new Scanner(System.in);

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

int n1=s.nextInt();

ThisDemo td=new ThisDemo();

td.m2();

td.setValues(n1);

td.display();

break;

case 3:

FinalDemo fd=new FinalDemo();

break;

case 4:

System.out.println("existing the program bye..!");

System.exit(0);

break;

default:

System.out.println("please choose valid choice");

break;

}

}

}

}

class A

{

int i=10;

public void m1()

{

System.out.println("I am in class A1");

}

A()

{

System.out.println("I am in class A2");

}

}

class B extends A

{

int i=20;

public void show()

{

System.out.println(super.i);

}

public void m1()

{

System.out.println("i am in class B1");

}

public void display()

{

m1();

super.m1();

System.out.println("i am in display method in B1");

}

B()

{

super();

System.out.println("i am in class B2");

}

}

class ThisDemo

{

int i;

void setValues(int i)

{

this.i=i;

}

void show()

{

System.out.println(i);

}

void display()

{

this.show();

System.out.println("welcome to the ThisDemo class");

}

ThisDemo()

{

this(10);

System.out.println("no argument constructor");

}

ThisDemo(int a)

{

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

}

void m1(ThisDemo td)

{

System.out.println("i am in m1 method");

}

void m2()

{

m1(this);

}

}

class FinalDemo

{

final int speed;

FinalDemo()

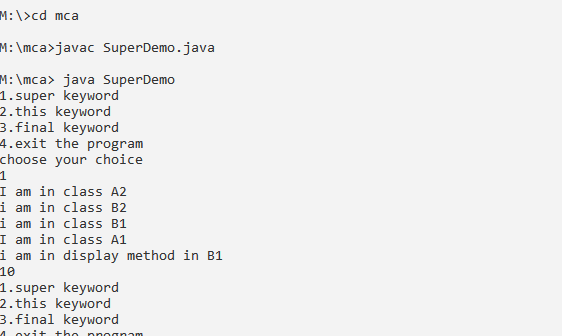
{

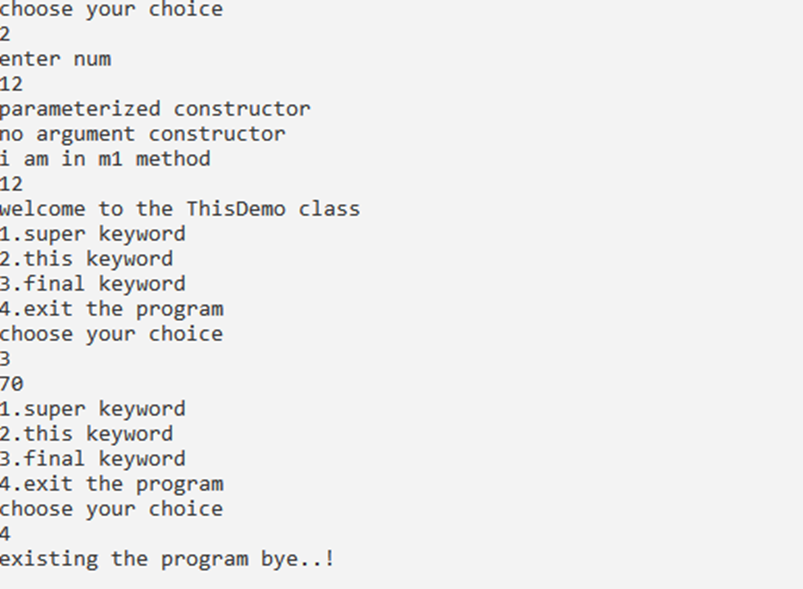
speed=70;

System.out.println(speed);

}}

OUTPUT:





import Mypack1.Demo111;

import Mypack1.\*;

interface Sum

{

public void Sum(int a,int b);

}

interface mul

{

public void mult(int a,int b,int c);

}

interface Divi extends Sum,mul

{

public void divi1(int a,int b);

}

class Rohit88 implements Divi

{

public void Sum(int a,int b)

{

int c=a+b;

System.out.println("sum of "+a+" and "+b+" is:"+c);

}

public void mult(int a,int b,int c)

{

int mul=a\*b\*c;

System.out.println("mul of "+a+","+b+" and "+c+"is:"+mul);

}

public void divi1(int a,int b)

{

int div=a%b;

System.out.println("divi "+a+" and "+b+" is: "+div);

}

public static void main(String args[])

{

Demo111 d1=new Demo111();

d1.one();

Rohit88 s=new Rohit88();

s.Sum(2,3);

s.mult(2,3,4);

s.divi1(2,4);

}

}

package Mypack1;

public class Demo111

{

public void one()

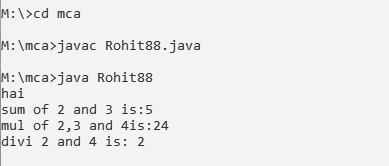
{

System.out.println("hai");

}

}

OUTPUT:



import java.time.LocalDateTime;

import java.util.Calendar;

import java.util.Scanner;

import java.util.\*;

public class Types

{

public static void main(String args[])

{

menu();

}

static void menu()

{

Scanner scn=new Scanner(System.in);

while(True)

{

System.out.println("1.Wrapper class");

System.out.println("2.Number class");

System.out.println("3.Date and Time class");

System.out.println("4.Calender class");

{ System.out.println("5.vector class");

} System.out.println("6.Stack class");

System.out.println("7.Exit the program");

System.out.println("choose your program");

int choice=scn.nextInt();

switch(choice)

{

case 1:

Integer a=new Integer(10);

int b=a;

int c=a.intValue();

System.out.println(a+""+b+""+c);

break;

case 2:

NUmber b1=1234567;

System.out.println(b1.byteValue());

break;

case 3:

LocalDateTime d=LocalDateTime.now();

System.out.println(d);

break;

case 4:

Calendar cal=Calendar.getInstantce();

System.out.println(cal.getTime());

cal.add(Calendar.YEAR,10);

System.out.println("5 months after"+""+cal.getTime());

cal.add(Calendar.DATE,-15);

System.out.println("5 days before"+""+cal.getTime());

break;

case 5:

Vector v=new Vector(5);

v.add("java");

v.add(10.75);

v.add(10);

v.add(1234.56);

System.out.println("Display the vector elements"+v);

System.out.println("Display the vector elements"+v.get(3));

System.out.println("Display the vector elements"+v.size());

System.out.println("Display the vector elements"+v.remove(3));

break;

case 6:

Stack st= new Stack();

st.push(10);

st.push(20);

System.out.println("Display the elements"+st);

st.pop();

System.out.println("Display the elements"+st);

System.out.println(st.peek());

break;

case 7:

System.out.println("Existing the program bye..!");

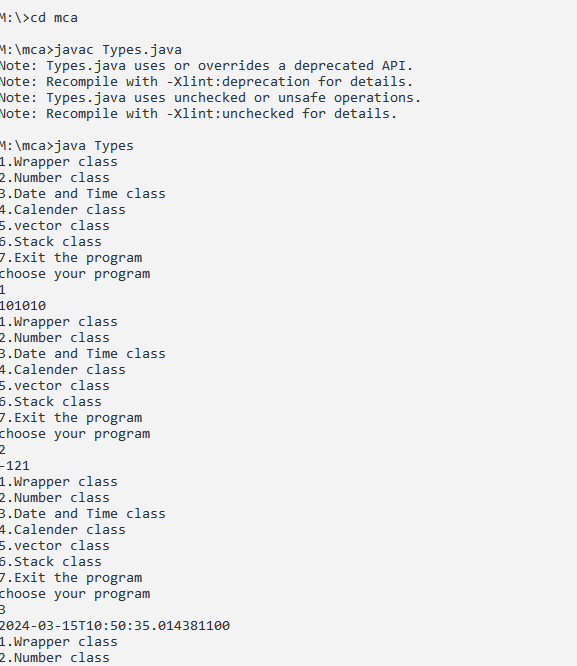
System.exit(0);

break;

default:

System.out.println("please choose valid choice");

break;



import java.util.Scanner;

public class Stringfun

{

public static void main(String args[])

{

menu();

}

static void menu()

{

Scanner scn=new Scanner(System.in);

while(true)

{

System.out.println("shf menu:");

System.out.println("1.charAT()-> it returns a char at specified position");

System.out.println("2.rquals-> it compares the two given strings and rerurn a boolean,that is T or F");

System.out.println("3.length()-> returns the length of a specified string");

System.out.println("4.tolower Case-> converts the string to lowercase letters");

System.out.println("5.toupper Case-> converts the string to upper case letters");

System.out.println("6.Indexof()->returns the first found position of a character");

System.out.println("7.substring()-> extracts the substring based on index values");

System.out.println("8.trim()-> removes the whitespace from both ends of string");

System.out.println("9.concat()-> appends one string to the end of another");

System.out.println("10.exit the progam");

System.out.println("choose your choice");

int choice=scn.nextInt();

switch(choice)

{

case 1:

String str="Hello";

System.out.println("Ch aracter at given position is:"+str.charAt(1));

break;

case 2:

String s1="java";

String s2="java";

String s3="java";

System.out.println("Compare s1 and s2:"+s1.equals(s2));

System.out.println("Compare s1 and s3:"+s1.equals(s3));

break;

case 3:

String s4="India";

System.out.println("The length os string is"+s4.length());

break;

case 4:

String s5="MCA";

System.out.println(s5.toLowerCase());

break;

case 5:

String s6="andhra pradesh";

System.out.println(s6.toUpperCase());

break;

case 6:

String s7="College";

System.out.println("Char-> 'g' position is"+s7.indexOf('g'));

break;

case 7:

String s8="MOTHER THERESA";

System.out.println(s8.substring(0,6));

System.out.println(s8.substring(3));

break;

case 8 :

String s9="Good morning";

System.out.println(s9.trim());

break;

case 9 :

String str1="Good";

System.out.println(str1.concat("Afternoon"));

break;

case 10:

System.out.println("Exiting from the menu bye...!");

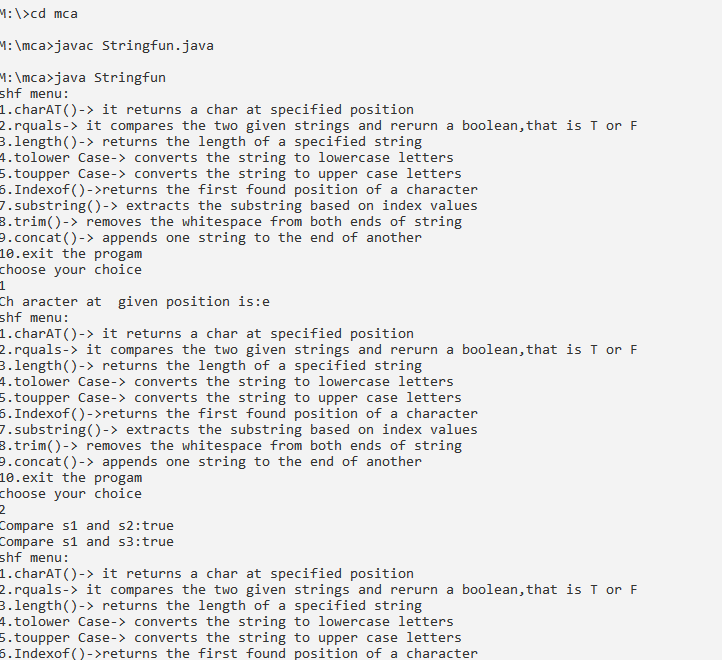
System.exit(0);

break;

default:

System.out.println("Option Invalid,please select valid option");

break;



import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

class Notepad extends Frame

{

Notepad()

{

Frame F=new Frame("Notepad");

MenuBar mb=new MenuBar();

Menu menu1=new Menu("File");

Menu menu2=new Menu("Edit");

Menu menu3=new Menu("Format");

Menu menu4=new Menu("View");

Menu menu5=new Menu("Help");

MenuItem i1=new MenuItem("New");

MenuItem i2=new MenuItem("New window");

MenuItem i3=new MenuItem("open");

MenuItem i4=new MenuItem("save");

MenuItem i5=new MenuItem("save as");

MenuItem i6=new MenuItem("page setup");

MenuItem i7=new MenuItem("print");

MenuItem i8=new MenuItem("Exit");

MenuItem j1=new MenuItem("undo");

MenuItem j2=new MenuItem("cut");

MenuItem j3=new MenuItem("copy");

MenuItem j4=new MenuItem("paste");

MenuItem j5=new MenuItem("Delete");

MenuItem j6=new MenuItem("Searech with bing");

MenuItem j7=new MenuItem("Find");

MenuItem j8=new MenuItem("Find Next");

MenuItem j9=new MenuItem("Find previous");

MenuItem j10=new MenuItem("Replace");

MenuItem j11=new MenuItem("Go To");

MenuItem j12=new MenuItem("Select All");

MenuItem j13=new MenuItem("Time/Date");

MenuItem k1=new MenuItem("word wrap");

MenuItem k2=new MenuItem("Font");

MenuItem p1=new MenuItem("Zoom");

MenuItem p2=new MenuItem("Status Bar");

MenuItem q1=new MenuItem("View help");

MenuItem q2=new MenuItem("Send FeedBack");

MenuItem q3=new MenuItem("About Notepad");

menu1.add(i1);

menu1.add(i2);

menu1.add(i3);

menu1.add(i4);

menu1.add(i5);

menu1.add(i6);

menu1.add(i7);

menu1.add(i8);

menu2.add(j1);

menu2.add(j2);

menu2.add(j3);

menu2.add(j4);

menu2.add(j5);

menu2.add(j6);

menu2.add(j7);

menu2.add(j8);

menu2.add(j9);

menu2.add(j10);

menu2.add(j11);

menu2.add(j12);

menu2.add(j13);

menu3.add(k1);

menu3.add(k2);

menu4.add(p1);

menu4.add(p2);

menu5.add(q1);

menu5.add(q2);

menu5.add(q3);

mb.add(menu1);

mb.add(menu2);

mb.add(menu3);

mb.add(menu4);

mb.add(menu5);

F.setMenuBar(mb);

F.setSize(400,400);

F.setLayout(null);

F.setVisible(true);

}

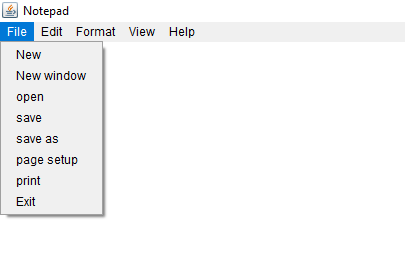
public static void main(String args[])

{

new Notepad();

}

}



import java.awt.\*;

import java.awt.event.\*;

public class RegistrationForm extends Frame {

Label titleLabel, nameLabel, genderLabel, aadharLabel1, aadharLabel2, motherLabel, fatherLabel, occupationLabel, dobLabel, bloodGroupLabel, nationalityLabel, addressLabel, telephoneLabel, mobileLabel, emailLabel, educationLabel;

TextField nameField, aadharField1, aadharField2, motherField, fatherField, occupationField, dobField, bloodGroupField, nationalityField, telephoneField, mobileField, emailField, casteField;

Choice genderChoice, categoryChoice;

TextArea addressArea, educationArea;

Button submitButton, cancelButton;

public RegistrationForm() {

setLayout(null);

setBackground(Color.LIGHT\_GRAY);

Font titleFont = new Font("arial", Font.BOLD, 25);

titleLabel = new Label("REGISTRATION FORM");

titleLabel.setBounds(200, 30, 400, 40);

titleLabel.setForeground(Color.BLACK);

titleLabel.setFont(titleFont);

nameLabel = new Label("1. NAME OF THE CANDIDATE (CAPITALS. As per SSC)");

nameLabel.setBounds(50, 80, 300, 20);

nameField = new TextField(10);

nameField.setBounds(400, 80, 250, 20);

genderLabel = new Label("2. GENDER");

genderLabel.setBounds(50, 110, 300, 20);

genderChoice = new Choice();

genderChoice.add("MALE");

genderChoice.add("FEMALE");

genderChoice.setBounds(400, 110, 250, 20);

aadharLabel1 = new Label("3. AADHAAR NUMBER");

aadharLabel1.setBounds(50, 140, 300, 20);

aadharField1 = new TextField(10);

aadharField1.setBounds(400, 140, 250, 20);

motherLabel = new Label("4. NAME OF THE MOTHER");

motherLabel.setBounds(50, 170, 300, 20);

motherField = new TextField(10);

motherField.setBounds(400, 170, 250, 20);

aadharLabel2 = new Label("5. AADHAAR NUMBER");

aadharLabel2.setBounds(50, 200, 300, 20);

aadharField2 = new TextField(10);

aadharField2.setBounds(400, 200, 250, 20);

fatherLabel = new Label("6. NAME OF THE FATHER/GUARDIAN");

fatherLabel.setBounds(50, 230, 300, 20);

fatherField = new TextField(10);

fatherField.setBounds(400, 230, 250, 20);

occupationLabel = new Label("7. OCCUPATION & ANNUAL INCOME OF THE PARENT");

occupationLabel.setBounds(50, 260, 300, 20);

occupationField = new TextField(10);

occupationField.setBounds(400, 260, 250, 20);

dobLabel = new Label("8. DATE OF BIRTH DD/MM/YYYY");

dobLabel.setBounds(50, 290, 300, 20);

dobField = new TextField(10);

dobField.setBounds(400, 290, 250, 20);

bloodGroupLabel = new Label("9. BLOOD GROUP");

bloodGroupLabel.setBounds(50, 320, 300, 20);

bloodGroupField = new TextField(10);

bloodGroupField.setBounds(400, 320, 250, 20);

nationalityLabel = new Label("10. NATIONALITY & RELIGION");

nationalityLabel.setBounds(50, 350, 300, 20);

nationalityField = new TextField(10);

nationalityField.setBounds(400, 350, 250, 20);

Label categoryLabel = new Label("11. CATEGORY OF RESERVATION");

categoryLabel.setBounds(50, 380, 300, 20);

categoryChoice = new Choice();

categoryChoice.add("OC");

categoryChoice.add("SC");

categoryChoice.add("ST");

categoryChoice.add("BC-A");

categoryChoice.add("BC-B");

categoryChoice.add("BC-C");

categoryChoice.add("BC-D");

categoryChoice.add("BC-E");

categoryChoice.setBounds(400, 380, 250, 20);

Label casteLabel = new Label("CASTE:");

casteLabel.setBounds(50, 410, 300, 20);

casteField = new TextField(10);

casteField.setBounds(400, 410, 250, 20);

addressLabel = new Label("12. ADDRESS FOR COMMUNICATION");

addressLabel.setBounds(50, 440, 300, 20);

addressArea = new TextArea(5, 10);

addressArea.setBounds(400, 440, 250, 90);

telephoneLabel = new Label("13. TELEPHONE NO.");

telephoneLabel.setBounds(50, 550, 300, 20);

telephoneField = new TextField(10);

telephoneField.setBounds(400, 550, 250, 20);

mobileLabel = new Label("14. MOBILE NUMBERS");

mobileLabel.setBounds(50, 580, 300, 20);

mobileField = new TextField(10);

mobileField.setBounds(400, 580, 250, 20);

emailLabel = new Label("15. E-MAIL ID");

emailLabel.setBounds(50, 610, 300, 20);

emailField = new TextField(10);

emailField.setBounds(400, 610, 250, 20);

educationLabel = new Label("16. EDUCATIONAL QUALIFICATIONS");

educationLabel.setBounds(50, 640, 300, 20);

educationArea = new TextArea(5, 10);

educationArea.setBounds(400, 640, 250, 90);

submitButton = new Button("Submit");

submitButton.setBounds(220, 750, 80, 30);

submitButton.setBackground(Color.BLACK);

submitButton.setForeground(Color.WHITE);

cancelButton = new Button("Cancel");

cancelButton.setBounds(320, 750, 80, 30);

cancelButton.setBackground(Color.BLACK);

cancelButton.setForeground(Color.WHITE);

add(titleLabel);

add(nameLabel);

add(nameField);

add(genderLabel);

add(genderChoice);

add(aadharLabel1);

add(aadharField1);

add(motherLabel);

add(motherField);

add(aadharLabel2);

add(aadharField2);

add(fatherLabel);

add(fatherField);

add(occupationLabel);

add(occupationField);

add(dobLabel);

add(dobField);

add(bloodGroupLabel);

add(bloodGroupField);

add(nationalityLabel);

add(nationalityField);

add(categoryLabel);

add(categoryChoice);

add(casteLabel);

add(casteField);

add(addressLabel);

add(addressArea);

add(telephoneLabel);

add(telephoneField);

add(mobileLabel);

add(mobileField);

add(emailLabel);

add(emailField);

add(educationLabel);

add(educationArea);

add(submitButton);

add(cancelButton);

setTitle("MOTHER THERESA INSTITUTE OF COMPUTER APPLICATIONS");

setSize(700, 800);

setVisible(true);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent windowEvent) {

System.exit(0);

}

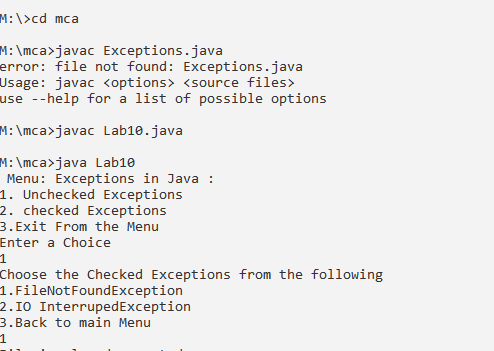
});

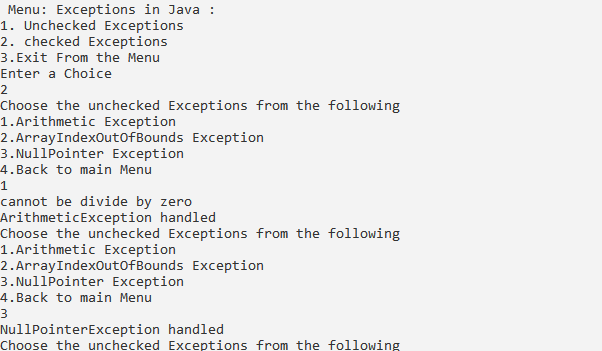
}

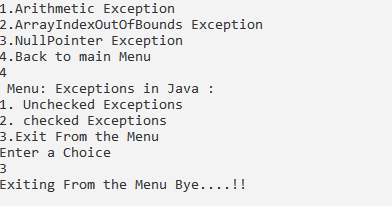
public static void main(String[] args) {

new RegistrationForm();

OUTPUT:







import java.lang.String;

import java.sql.\*;

class Oracleconn

{

public static void main(String[] args)

{

try

{

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@10.199.20.3:1521:mother","mtcab2388","mtcab2388");

Statement stmt=con.createStatement();

stmt.execute("create table stu88(roll number(3),name varchar(20),marks number(3))");

System.out.println("table created successfully");

stmt.executeUpdate("insert into stu88 values(1,'rohitkumar',88)");

System.out.println("one record inserted successfuly");

}

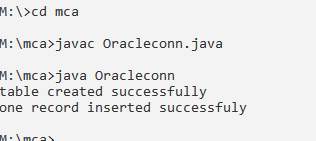
}

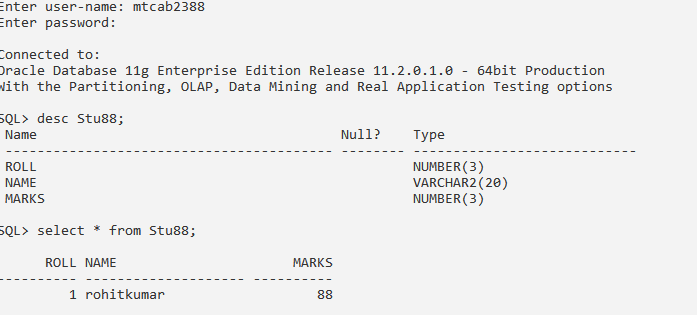
catch (Exception e)

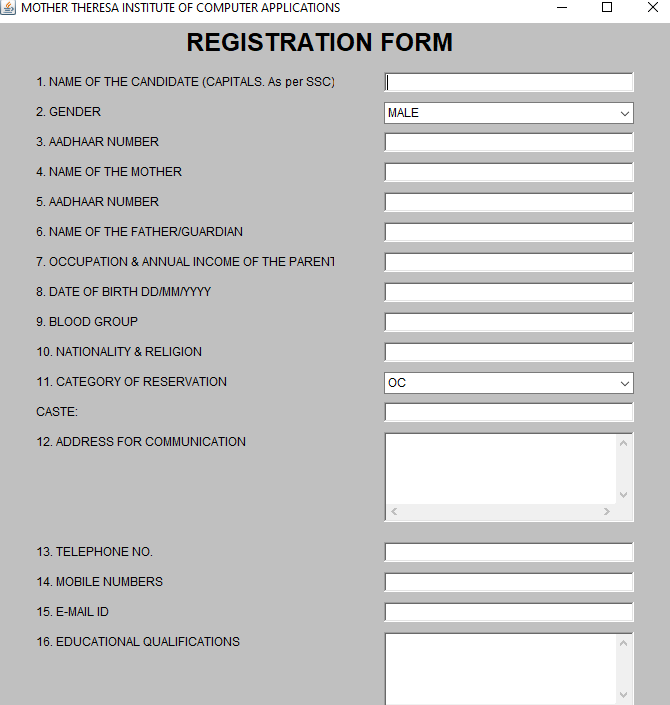
{

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

OUTPUT:







import java.awt.\*;

import java.awt.Color;

class LoginDemo extends Frame

{

TextField tf1,tf2;

Label l1,l2;

Button b1,b2;

public LoginDemo()

{

Frame F=new Frame("Login Form");

F.setTitle("Login Form");

tf1=new TextField();

tf2=new TextField();

l1= new Label("Username");

l2=new Label("password");

b1=new Button("sign in");

b2=new Button("sign out");

l1.setBounds(30,100,100,50);

tf1.setBounds(140,100,50,50);

l2.setBounds(30,160,100,50);

tf2.setBounds(75,250,100,50);

b1.setBounds(75,250,100,50);

b2.setBounds(200,250,100,50);

F.add(l1);

F.add(tf1);

F.add(l2);

F.add(tf2);

F.add(b2);

F.setSize(500,500);

F.setLayout(null);

F.setVisible(true);

F.setBackground(Color.GREEN);

}

public static void main(String args[])

{

new LoginDemo();

}

}

/\*

<applet code="LoginDemo.class" width="500" height="500">

</applet

**OUTPUT:**



import java.io.FileNotFoundException;

//import java.io.FileOutputStream;

import java.util.Scanner;

import java.io.File;

class Exceptions

{

public static void main(String[] args)

{

menu();

}

static void menu()

{

Scanner scn = new Scanner(System.in);

while(true)

{

System.out.println(" Menu: Exceptions in Java : ");

System.out.println("1. Unchecked Exceptions");

System.out.println("2. checked Exceptions");

System.out.println("3.Exit From the Menu");

System.out.println("Enter a Choice");

int choice = scn.nextInt();

switch(choice)

{

case 1:

unchecked();

break;

case 2:

checked();

break;

case 3:

System.out.println("Exiting From the Menu Bye....!!");

System.exit(0);

break;

default:

System.out.println("Option Invalid,Please Select Valid Option");

break;

}

}

}

static void unchecked()

{

Scanner scn=new Scanner(System.in);

while(true)

{

System.out.println("Choose the Checked Exceptions from the following");

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

System.out.println("2.IO InterrupedException");

System.out.println("3.Back to main Menu");

int ch = scn.nextInt();

switch(ch)

{

case 1:

try{

File f=new File("M:\\Java.txt\\");

//FileOutputStream f=new FileOutputStream("M:\\Java.txt\\");

Scanner s=new Scanner(f);

String x=s.nextLine();

System.out.println("File is Created Sucessfully"+x);

}

catch(FileNotFoundException e)

{

System.out.println("File is already created");

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

}

break;

case 2:

try

{

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

{

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

Thread.sleep(2000);

}

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

}

catch (InterruptedException e)

{

System.out.println(e);

}

break;

case 3:

menu();

break;

default:

System.out.println("Select Valid Option");

break;

}

}

}

static void checked()

{

Scanner scn=new Scanner(System.in);

while(true)

{

System.out.println("Choose the unchecked Exceptions from the following");

System.out.println("1.Arithmetic Exception");

System.out.println("2.ArrayIndexOutOfBounds Exception");

System.out.println("3.NullPointer Exception");

System.out.println("4.Back to main Menu");

int ch = scn.nextInt();

switch(ch)

{

case 1:

try{

int a=10;

System.out.println(a/0);

}

catch(ArithmeticException e)

{

System.out.println("cannot be divide by zero");

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

}

break;

case 2:

try

{

int[]a={10,20,30};

System.out.println(a[10]);

}

catch (ArrayIndexOutOfBoundsException e)

{

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

}

break;

case 3:

try

{

int[] s=null;

System.out.println(s[0]);

}

catch (NullPointerException e)

{

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

}

break;

case 4:

menu();

break;

default:

System.out.println("Select Valid Option");

break;

}}}}

import java.util.Scanner;

public class AbTypes

{

public static void main(String []args ){

menu();

}static void menu(){

Scanner scn=new Scanner(System.in);

while(true){

System.out.println("MENU");

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

System.out.println("2.NESTED OR INNER CLASS");

System.out.println("3.EXIT");

int choice=scn.nextInt();

switch(choice){

case 1:

B b=new B();

b.m1();

b.m2();

b.m3();

break;

case 2:

Scanner sc =new Scanner(System.in);

Out o=new Out();

Out.Inner i=o.new Inner();

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

int n1=sc.nextInt();

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

int n2=sc.nextInt();

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

int n3=sc.nextInt();

o.sum(n1,n2);

o.mul(n1,n2,n3);

i.m1();

break;

case 3:

System.out.println("EXIT THE PROGRAM");

System.exit(0);

break;

default:

System.out.println("CHOOSE VALID CHOICE");

break;

}}}}

abstract class A {

abstract void m1();

public void m2(){

System.out.println("m2() defined in class A");

}

public void m3(){

System.out.println("m3() defined in class A");

}

}

class B extends A{

@Override

public void m1(){

System.out.println("ABSTRACT METHOD M1() DEFINED IN CLASS B");

}public void m2(){

System.out.println("DEFAULT METHOD M2() DEFINED IN CLASS B");

}

}class Out{

public void sum(int a,int b){

int c=a+b;

System.out.println("SUM OF "+a+"and"+b+"is:"+c);

}

public void mul(int a,int b,int c){

int d=a\*b\*c;

System.out.println("MUL OF "+a+"and"+b+"and"+c+"is:"+d);

}

class Inner{

public void m1(){

System.out.println("M1() IS INNER CLASS OF THE PROGRAM");

}

}

}

}

EXCEPTIONS

import java.io.FileNotFoundException;

//import java.io.FileOutputStream;

import java.util.Scanner;

import java.io.File;

class Exceptions

{

public static void main(String[] args)

{

menu();

}

static void menu()

{

Scanner scn = new Scanner(System.in);

while(true)

{

System.out.println(" Menu: Exceptions in Java : ");

System.out.println("1. Unchecked Exceptions");

System.out.println("2. checked Exceptions");

System.out.println("3.Exit From the Menu");

System.out.println("Enter a Choice");

int choice = scn.nextInt();

switch(choice)

{

case 1:

unchecked();

break;

case 2:

checked();

break;

case 3:

System.out.println("Exiting From the Menu Bye....!!");

System.exit(0);

break;

default:

System.out.println("Option Invalid,Please Select Valid Option");

break;

}

}

}

static void unchecked()

{

Scanner scn=new Scanner(System.in);

while(true)

{

System.out.println("Choose the Checked Exceptions from the following");

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

System.out.println("2.IO InterrupedException");

System.out.println("3.Back to main Menu");

int ch = scn.nextInt();

switch(ch)

{

case 1:

try{

File f=new File("M:\\Java.txt\\");

//FileOutputStream f=new FileOutputStream("M:\\Java.txt\\");

Scanner s=new Scanner(f);

String x=s.nextLine();

System.out.println("File is Created Sucessfully"+x);

}

catch(FileNotFoundException e)

{

System.out.println("File is already created");

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

}

break;

case 2:

try

{

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

{

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

Thread.sleep(2000);

}

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

}

catch (InterruptedException e)

{

System.out.println(e);

}

break;

case 3:

menu();

break;

default:

System.out.println("Select Valid Option");

break;

}

}

}

static void checked()

{

Scanner scn=new Scanner(System.in);

while(true)

{

System.out.println("Choose the unchecked Exceptions from the following");

System.out.println("1.Arithmetic Exception");

System.out.println("2.ArrayIndexOutOfBounds Exception");

System.out.println("3.NullPointer Exception");

System.out.println("4.Back to main Menu");

int ch = scn.nextInt();

switch(ch)

{

case 1:

try{

int a=10;

System.out.println(a/0);

}

catch(ArithmeticException e)

{

System.out.println("cannot be divide by zero");

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

}

break;

case 2:

try

{

int[]a={10,20,30};

System.out.println(a[10]);

}

catch (ArrayIndexOutOfBoundsException e)

{

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

}

break;

case 3:

try

{

int[] s=null;

System.out.println(s[0]);

}

catch (NullPointerException e)

{

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

}

break;

case 4:

menu();

break;

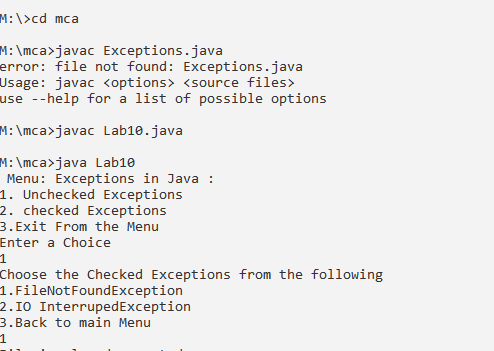
default:

System.out.println("Select Valid Option");

break;

}}}}

Output:



import java.io.\*;

import java.util.\*;

class PageReplace1{

static boolean search(int key,int[] fr)

{

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

if(fr[i]==key)

return true;

return false;

}

static int predict(int pg[],int[] fr,int pn,int index)

{

int res=-1,farthest=index;

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

int j;

for(j=index;j<pn;j++){

if(fr[i]==pg[j]){

if(j>farthest){

farthest=j;

res=i;

}

break;

}

}

if(j==pn)

return i;

}

return(res==-1)?0:res;

}

static void optimalPage(int pg[],int pn,int fn)

{

int[] fr=new int[fn];

int hit=0;

int index=0;

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

if(search(pg[i],fr)){

hit++;

continue;

}

if(index < fn)

fr[index++]=pg[i];

else {

int j=predict(pg,fr,pn,i+1);

fr[j]=pg[i];

}

}

System.out.println("No.of hits="+hit);

System.out.println("No.of misses="+(pn-hit));

}

public static void main(String[] args)

{

int pg[]

={7,0,1,2,0,3,0,4,2,3,0,3,2};

int pn=pg.length;

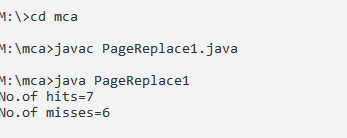
int fn=4;

optimalPage(pg,pn,fn);

}

}

OUTPUT:



import java.util.Scanner;

class InsertDelete

{

public static void main(String[] args)

{

menu();

}

static void menu()

{

Scanner sc=new Scanner(System.in);

while(true)

{

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

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

System.out.println("3.exit");

System.out.println("select any option");

int choice=sc.nextInt();

switch(choice)

{

case 1:

int len,pos,e,i;

Scanner s=new Scanner(System.in);

System.out.println("enter number of elements in an array");

len=s.nextInt();

int arr[]=new int[len+1];

System.out.println("enter elements of an arry");

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

{

arr[i]=s.nextInt();

}

System.out.println("enter position where to insert an element");

pos=s.nextInt();

System.out.println("enter element to be inserted");

e=s.nextInt();

for(i=len-1;i>=(pos-1);i--)

{

arr[i+1]=arr[i];

}

arr[pos-1]=e;

System.out.println("after inserting array elements are:");

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

{

System.out.println(arr[i]+" ");

}

break;

case 2:

int len1,pos1,e1,i1;

Scanner s1=new Scanner(System.in);

System.out.println("enter number of elements in an array");

len1=s1.nextInt();

int arr1[]=new int[len1];

System.out.println("enter elements of an arry");

for(i1=0;i1<len1;i1++)

{

arr1[i1]=s1.nextInt();

}

System.out.println("enter element to be deleted");

e1=s1.nextInt();

for(i1=0;i1<len1-1;i1++)

{

if(arr1[i1]==e1)

{

for(int j=i1;j<(len1-1);j++)

{

arr1[j]=arr1[j+1];

}

break;

}

}

System.out.println("after deleting array elements are:");

for(i1=0;i1<(len1-1);i1++)

{

System.out.println(arr1[i1]);

}

break;

case 3:

System.out.println("exiting the program");

System.exit(0);

break;

default:

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

break;

}

}

}

}

OUTPUT:

