**Career College, Bhopal**

**Department of Computer Science**

**Session 2014-15**

**M.Sc.(CS)-II Sem**

**JAVA**

* Write a program to find the greatest number among three given numbers using conditional operator.
* Write a program to display: 1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

* Write a program to demonstrate a two dimensional array with row sum.
* Write a program to demonstrate constructor overloading.
* Design a class to represent a bank account. Include the following members-
* Data members
* Name of the depositor
* Account number
* Type of account
* Balance amount in the account
* Methods
* To assign initial values
* To deposit an amount
* To withdraw an amount after checking balance
* To display the name and balance
* Write a program to demonstrate multilevel inheritance.
* Give an example where interface can be used to support Multiple Inheritance.
* Write a program to construct multiplication table of 3 using multithreading.
* Write a program to demonstrate the try, catch and finally block in exception handling.
* Develop an applet that receive three numeric values as input from the user and then displays the largest of the three on the screen. Write an HTML page and test the applet

**solutions**

**import java.util.Scanner;**

**public class JavaExample**

**{**

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

**{**

**int num1, num2, num3, result, temp;**

**/\* Scanner is used for getting user input.**

**\* The nextInt() method of scanner reads the**

**\* integer entered by user.**

**\*/**

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

**System.out.println("Enter First Number:");**

**num1 = scanner.nextInt();**

**System.out.println("Enter Second Number:");**

**num2 = scanner.nextInt();**

**System.out.println("Enter Third Number:");**

**num3 = scanner.nextInt();**

**scanner.close();**

**/\* In first step we are comparing only num1 and**

**\* num2 and storing the largest number into the**

**\* temp variable and then comparing the temp and**

**\* num3 to get final result.**

**\*/**

**temp = num1>num2 ? num1:num2;**

**result = num3>temp ? num3:temp;**

**System.out.println("Largest Number is:"+result);**

**}**

**}**

**Output:**

**Enter First Number:**

**89**

**Enter Second Number:**

**109**

**Enter Third Number:**

**8**

**Largest Number is:109**

**Ans.2**

**public class Pattern {**

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

**char last = 'E', alphabet = 'A';**

**for(int i = 1; i <= (last-'A'+1); ++i) {**

**for(int j = 1; j <= i; ++j) {**

**System.out.print(alphabet + " ");**

**}**

**++alphabet;**

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

**}**

**}**

**}**

**output:**

A

B B

C C C

D D D D

E E E E E

**Ans.3**

**class Lab4 {**

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

**int [][] scores = {{ 20, 18, 23, 20, 16 },**

**{ 30, 20, 18, 21, 20 },**

**{ 16, 19, 16, 53, 24 },**

**{ 25, 24, 22, 24, 25 }};**

**outputArray(scores);**

**}**

**public static void outputArray(int[][] array) {**

**int sum= 0;**

**int rowSize = array.length;**

**int columnSize = array[0].length;**

**System.out.println("rows=" + rowSize + "cols=" + columnSize);**

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

**for (int j = 0; j < array[0].length; j++) {**

**sum += array[i][j];**

**}**

**System.out.println("Print the sum of rows = " + sum);**

**}**

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

**sum = 0;**

**sum = sum + array[i][j];**

**// It is telling me the j can't be resolved**

**}**

**}**

**}**

**The program prints out:**

**rows=4cols=5**

**Print the sum of rows = 612**

**Print the sum of rows = 20358**

**Print the sum of rows = 652058**

**Print the sum of rows = 20866609**

**Ans 4.**

class StudentData

{

private int stuID;

private String stuName;

private int stuAge;

StudentData()

{

//Default constructor

stuID = 100;

stuName = "New Student";

stuAge = 18;

}

StudentData(int num1, String str, int num2)

{

//Parameterized constructor

stuID = num1;

stuName = str;

stuAge = num2;

}

//Getter and setter methods

public int getStuID() {

return stuID;

}

public void setStuID(int stuID) {

this.stuID = stuID;

}

public String getStuName() {

return stuName;

}

public void setStuName(String stuName) {

this.stuName = stuName;

}

public int getStuAge() {

return stuAge;

}

public void setStuAge(int stuAge) {

this.stuAge = stuAge;

}

public static void main(String args[])

{

//This object creation would call the default constructor

StudentData myobj = new StudentData();

System.out.println("Student Name is: "+myobj.getStuName());

System.out.println("Student Age is: "+myobj.getStuAge());

System.out.println("Student ID is: "+myobj.getStuID());

/\*This object creation would call the parameterized

\* constructor StudentData(int, String, int)\*/

StudentData myobj2 = new StudentData(555, "Chaitanya", 25);

System.out.println("Student Name is: "+myobj2.getStuName());

System.out.println("Student Age is: "+myobj2.getStuAge());

System.out.println("Student ID is: "+myobj2.getStuID());

}

}

Output:

Student Name is: New Student

Student Age is: 18

Student ID is: 100

Student Name is: Chaitanya

Student Age is: 25

Student ID is: 555

Ans.5

class BankWork

{

final int max\_limit=20;

final int min\_limit=1;

final double min\_bal=500;

private String name[]=new String[20];

privateint accNo[]=newint[20];

private String accType[]=new String[20];

privatedouble balAmt[]=newdouble[20];

staticint totRec=0;

//constructor

BankWork()

{

for(int i=0;i<max\_limit;i++)

{

name[i]="";

accNo[i]=0;

accType[i]="";

balAmt[i]=0.0;

}

}

//TO ADD NEW RECORDpublicvoid newEntry()

{

String str;

int acno;

double amt;

boolean permit;

permit=true;

if (totRec>max\_limit)

{

System.out.println("\n\n\nSorry we cannot admit you in our bank...\n\n\n");

permit=false;

}

if(permit = true) //Allows to create new entry

{

totRec++; // Incrementing Total Record

System.out.println("\n\n\n=====RECORDING NEW ENTRY=====");

try{

accNo[totRec]=totRec; //Created AutoNumber to accNo so no invalid id occurs

System.out.println("Account Number : "+accNo[totRec]);

BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));

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

System.out.flush();

name[totRec]=obj.readLine();

System.out.print("Enter Account Type : ");

System.out.flush();

accType[totRec]=obj.readLine();

do{

System.out.print("Enter Initial Amount to be deposited : ");

System.out.flush();

str=obj.readLine();

balAmt[totRec]=Double.parseDouble(str);

}while(balAmt[totRec]<min\_bal); //Validation that minimun amount must be 500

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

}

catch(Exception e)

{}

}

}

//TO DISPLAY DETAILS OF RECORDpublicvoid display()

{

String str;

int acno=0;

boolean valid=true;

System.out.println("\n\n=====DISPLAYING DETAILS OF CUSTOMER=====\n");

try{

BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter Account number : ");

System.out.flush();

str=obj.readLine();

acno=Integer.parseInt(str);

if (acno<min\_limit || acno>totRec) //To check whether accNo is valid or Not

{

System.out.println("\n\n\nInvalid Account Number \n\n");

valid=false;

}

if (valid==true)

{

System.out.println("\n\nAccount Number : "+accNo[acno]);

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

System.out.println("Account Type : "+accType[acno]);

System.out.println("Balance Amount : "+balAmt[acno]+"\n\n\n");

}

}

catch(Exception e)

{}

}

//TO DEPOSIT AN AMOUNTpublicvoid deposit()

{

String str;

double amt;

int acno;

boolean valid=true;

System.out.println("\n\n\n=====DEPOSIT AMOUNT=====");

try{

//Reading deposit value

BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter Account No : ");

System.out.flush();

str=obj.readLine();

acno=Integer.parseInt(str);

if (acno<min\_limit || acno>totRec) //To check whether accNo is valid or Not

{

System.out.println("\n\n\nInvalid Account Number \n\n");

valid=false;

}

if (valid==true)

{

System.out.print("Enter Amount you want to Deposit : ");

System.out.flush();

str=obj.readLine();

amt=Double.parseDouble(str);

balAmt[acno]=balAmt[acno]+amt;

//Displaying Depsit Details

System.out.println("\nAfter Updation...");

System.out.println("Account Number : "+acno);

System.out.println("Balance Amount : "+balAmt[acno]+"\n\n\n");

}

}

catch(Exception e)

{}

}

//TO WITHDRAW BALANCEpublicvoid withdraw()

{

String str;

double amt,checkamt;

int acno;

boolean valid=true;

System.out.println("\n\n\n=====WITHDRAW AMOUNT=====");

try{

//Reading deposit value

BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter Account No : ");

System.out.flush();

str=obj.readLine();

acno=Integer.parseInt(str);

if (acno<min\_limit || acno>totRec) //To check whether accNo is valid or Not

{

System.out.println("\n\n\nInvalid Account Number \n\n");

valid=false;

}

if (valid==true)

{

System.out.println("Balance is : "+balAmt[acno]);

System.out.print("Enter Amount you want to withdraw : ");

System.out.flush();

str=obj.readLine();

amt=Double.parseDouble(str);

checkamt=balAmt[acno]-amt;

if(checkamt >= min\_bal)

{

balAmt[acno]=checkamt;

//Displaying Depsit Details

System.out.println("\nAfter Updation...");

System.out.println("Account Number : "+acno);

System.out.println("Balance Amount : "+balAmt[acno]+"\n\n\n");

}

else

{

System.out.println("\n\nAs per Bank Rule you should maintain minimum balance of Rs 500\n\n\n");

}

}

}

catch(Exception e)

{}

}

};

class Bank

{

publicstaticvoid main(String args[])

{

String str;

int choice;

choice=0;

BankWork BW\_obj = new BankWork();

do

{

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

System.out.println("1) New Record Entry ");

System.out.println("2) Display Record Details ");

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

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

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

System.out.print("Enter your choice : ");

System.out.flush();

try{

BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));

str=obj.readLine();

choice=Integer.parseInt(str);

switch(choice)

{

case 1 : //New Record Entry

BW\_obj.newEntry();

break;

case 2 : //Displaying Record Details

BW\_obj.display();

break;

case 3 : //Deposit...

BW\_obj.deposit();

break;

case 4 : //Withdraw...

BW\_obj.withdraw();

break;

case 5 : System.out.println("\n\n.....THANKS FOR VISITING.....");

break;

default : System.out.println("\nInvalid Choice \n\n");

}

}

catch(Exception e)

{}

}while(choice!=5);

Ans5.

**class Car{**

**public Car()**

**{**

**System.out.println("Class Car");**

**}**

**public void vehicleType()**

**{**

**System.out.println("Vehicle Type: Car");**

**}**

**}**

**class Maruti extends Car{**

**public Maruti()**

**{**

**System.out.println("Class Maruti");**

**}**

**public void brand()**

**{**

**System.out.println("Brand: Maruti");**

**}**

**public void speed()**

**{**

**System.out.println("Max: 90Kmph");**

**}**

**}**

**public class Maruti800 extends Maruti{**

**public Maruti800()**

**{**

**System.out.println("Maruti Model: 800");**

**}**

**public void speed()**

**{**

**System.out.println("Max: 80Kmph");**

**}**

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

**{**

**Maruti800 obj=new Maruti800();**

**obj.vehicleType();**

**obj.brand();**

**obj.speed();**

**}**

**}**

**Output:**

**Class Car**

**Class Maruti**

**Maruti Model: 800**

**Vehicle Type: Car**

**Brand: Maruti**

**Max: 80Kmph**

Ans6.

interface vehicleone{

int speed=90;

public void distance();

}

interface vehicletwo{

int distance=100;

public void speed();

}

class Vehicle implements vehicleone,vehicletwo{

public void distance(){

int distance=speed\*100;

System.out.println("distance travelled is "+distance);

}

public void speed(){

int speed=distance/100;

}

}

class MultipleInheritanceUsingInterface{

public static void main(String args[]){

System.out.println("Vehicle");

obj.distance();

obj.speed();

}

}

Sample Output

Output is:

distance travelled is 9000

Ans7.

//Program: Multiplication Table Using Thread

public class Multiplicatin\_Table implements Runnable{

private int number;

public Multiplicatin\_Table(int number) {

this.number=number;

}

@Override

public void run() {

// TODO Auto-generated method stub

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

System.out.printf("%s: %d \* %d = %d\n", Thread.currentThread().getName(),

number, i, i \* number);

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("I will print table of 1 to 3 ");

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

Multiplicatin\_Table mul = new Multiplicatin\_Table(i);

Thread thread = new Thread(mul);

thread.start();

}

}

}

Output

c:\javaprg\Java programs\thread>javac Multiplicatin\_Table.java

c:\javaprg\Java programs\thread>java Multiplicatin\_Table

I will print table of 1 to 3

Thread-0: 1 \* 1 = 1

Thread-0: 1 \* 2 = 2

Thread-0: 1 \* 3 = 3

Thread-0: 1 \* 4 = 4

Thread-0: 1 \* 5 = 5

Thread-0: 1 \* 6 = 6

Thread-0: 1 \* 7 = 7

Thread-0: 1 \* 8 = 8

Thread-0: 1 \* 9 = 9

Thread-0: 1 \* 10 = 10

Thread-1: 2 \* 1 = 2

Thread-2: 3 \* 1 = 3

Thread-1: 2 \* 2 = 4

Thread-1: 2 \* 3 = 6

Thread-1: 2 \* 4 = 8

Thread-1: 2 \* 5 = 10

Thread-2: 3 \* 2 = 6

Thread-1: 2 \* 6 = 12

Thread-1: 2 \* 7 = 14

Thread-1: 2 \* 8 = 16

Thread-1: 2 \* 9 = 18

Thread-1: 2 \* 10 = 20

Thread-2: 3 \* 3 = 9

Thread-2: 3 \* 4 = 12

Thread-2: 3 \* 5 = 15

Thread-2: 3 \* 6 = 18

Thread-2: 3 \* 7 = 21

Thread-2: 3 \* 8 = 24

Thread-2: 3 \* 9 = 27

Thread-2: 3 \* 10 = 30

ANS 7.

class TestExceptions {

static void myMethod(int testnum) throws Exception {

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

if (testnum == 12)

throw new Exception();

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

return;

}

public static void main(String args[]) {

int testnum = 12;

try {

System.out.println("try - first statement");

myMethod(testnum);

System.out.println("try - last statement");

}

catch ( Exception ex) {

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

}

finally {

System. out. println( "finally") ;

}

System.out.println("Out of try/catch/finally - statement");

}

}

Output:

try - first statement

start - myMethod

An Exception

finally

Out of try/catch/finally - statement

ANS.

/\* <applet code="MaxOf3No" height=150 width=400> </applet> \*/

import java.awt.\*;

import java.applet.\*;

publicclass MaxOf3No extends Applet

{

TextField T1,T2,T3;

publicvoid init(){

T1 = new TextField(10);

T2 = new TextField(10);

T3 = new TextField(10);

add(T1);

add(T2);

add(T3);

T1.setText("0");

T2.setText("0");

T3.setText("0");

}

publicvoid paint(Graphics g){

int a, b, c,result;

String str;

g.drawString("Enter value to Check the Maximum of 3 ",10,50);

str=T1.getText();

a=Integer.parseInt(str);

str=T2.getText();

b=Integer.parseInt(str);

str=T3.getText();

c=Integer.parseInt(str);

g.setColor(Color.blue);

if (a>b) {

if (a>c)

result=a;

else

result=c;

}

else{

if (b>c)

result=b;

else

result=c;

}

g.drawString("Maximnum of 3 No is "+result,10,70);

showStatus("MAXIMUM OF 3 NUMBERS");

}

public boolean action(Event e, Object o){

repaint();

return true;

}

}