**Lab Exercise No:**29

**Exercise Objective(s):***Package*

**Exercise:***Create a package called shapes. Create some classes in the package representing some*

*common geometric shapes like Square, Triangle, Circle and so on. Create a class called*

*TestShapes and create objects for all the shapes and print corresponding messages.*

*Execute the TestShapes class.*

**Solution 29:**

**package** com.shape;

**public** **class** Square {

**void** itIsSquare(){

//function which is called by TestShapes

System.***out***.println("This ia a square");

}

}

**package** com.shape;

**public** **class** Triangle {

**void** itIsTriangle(){

//function which is called by TestShapes

System.***out***.println("This ia a Triangle");

}

}

**package** com.shape;

**public** **class** Circle {

**void** itIsCircle(){

//function which is called by TestShapes

System.***out***.println("This ia a Circle");

}

}

**package** com.shape;

**public** **class** TestShapes {

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

//creating the objects and calling the function of different classes

Square square=**new** Square();

square.itIsSquare();

//creating object for triangle class

Triangle triangle=**new** Triangle();

triangle.itIsTriangle();//calling function of class Triangle

//creating object for Circle class

Circle circle=**new** Circle();

circle.itIsCircle();//calling function of class Circle

}

**Lab Exercise No:**30

**Exercise Objective(s):***Jar*

**Exercise:**

1. *Create a new project in which create a package named org.animals. In that create various classes like Lion, Tiger, Deer, Monkey, Elephant and Giraffe. In each class create data members like color, weight,age etc. Create methods like isVegetarian, canClimb, sound etc*
2. *Create another project and in that create a package called zoo and create a class called VandalurZooand create objects for the animals that are existing in zoo and print the characteristic of each animal.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):***Export the jar and add it as an External Archive.*

**Solution 30:**

**package** org.animals;

**public** **class** Lion {

String color;

**int** weight,age;

**public** Lion(String color, **int** weight, **int** age) { //parameterized constructor

**super**();

System.***out***.println("This is a lion");

**this**.color = color;

**this**.weight = weight;

**this**.age = age;

}

**public** **void** getColor() {

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

}

**public** **void** getWeight() {

System.***out***.println(**this**.weight + " kgs.");

}

**public** **void** getAge() {

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

}

**public** **void** isVegetarian() {

System.***out***.println("Lion is not vegetarian");

}

**public** **void** canClimb() {

System.***out***.println("Lions cannot climb trees");

}

**public** **void** sound() {

System.***out***.println("Lion roars");

}

}

**package** org.animals;

**public** **class** Tiger {

String color;

**int** weight, age;

**public** Tiger(String color, **int** weight, **int** age) { //parameterized constructor

**super**();

System.***out***.println("This is a tiger");

**this**.color = color;

**this**.weight = weight;

**this**.age = age;

}

**public** **void** getColor() {

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

}

**public** **void** getWeight() {

System.***out***.println(**this**.weight + " kgs.");

}

**public** **void** getAge() {

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

}

**public** **void** isVegetarian() {

System.***out***.println("Tiger is not a vegetarian");

}

**public** **void** canClimb() {

System.***out***.println("Tiger cannot climb trees");

}

**public** **void** sound() {

System.***out***.println("Tigers growls");

}

}

**package** org.animals;

**public** **class** Deer {

String color;

**int** weight, age;

**public** Deer(String color, **int** weight, **int** age) { //parameterized constructor

**super**();

System.***out***.println("This is a deer");

**this**.color = color;

**this**.weight = weight;

**this**.age = age;

}

**public** **void** getColor() {

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

}

**public** **void** getWeight() {

System.***out***.println(**this**.weight + " kgs.");

}

**public** **void** getAge() {

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

}

**public** **void** isVegetarian() {

System.***out***.println("Deers are vegetarian");

}

**public** **void** canClimb() {

System.***out***.println("Deers cannot climb trees");

}

**public** **void** sound() {

System.***out***.println("Deer grunts");

}

}

**package** org.animals;

**import** org.animals.\*;

**public** **class** VandalurZooand {

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

Lion lion =**new** Lion("ABC",140,19);

System.***out***.print("Color is : ");

lion.getColor();

System.***out***.print("Age is : ");

lion.getAge();

System.***out***.print("Weight is : ");

lion.getWeight();

System.***out***.print("Capability to climb : ");

lion.canClimb();

System.***out***.print("Vegetarian or not : ");

lion.isVegetarian();

System.***out***.print("Animal sound : ");

lion.sound();

Tiger tiger =**new** Tiger("Orange",150,19);

System.***out***.print("Color is : ");

tiger.getColor();

System.***out***.print("Age is : ");

tiger.getAge();

System.***out***.print("Weight is : ");

tiger.getWeight();

System.***out***.print("Capability to climb : ");

tiger.canClimb();

System.***out***.print("Vegetarian or not : ");

tiger.isVegetarian();

System.***out***.print("Animal sound : ");

tiger.sound();

Deer deer =**new** Deer("Ochre",40,22);

System.***out***.print("Color is : ");

deer.getColor();

System.***out***.print("Age is : ");

deer.getAge();

System.***out***.print("Weight is : ");

deer.getWeight();

System.***out***.print("Capability to climb : ");

deer.canClimb();

System.***out***.print("Vegetarian or not : ");

deer.isVegetarian();

System.***out***.print("Animal sound : ");

deer.sound();

}

}

**Lab Exercise No:**31

**Exercise Objective(s):***System class*

**Exercise:***Create a class which displays the following about the JVM.*

1. *Version of Java*
2. *Vendor for Java*
3. *Class Path*
4. *Installed home directory*
5. *OS name on which it is installed with version*

Solution 31:

**package** com.day.four;

**public** **class** JVMDetails {

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

//Accessing from System class

System.***out***.println("Version is: "+System.*getProperty*("os.version"));

System.***out***.println("Running Java vm vendor name is: "+System.*getProperty*("java.vm.vendor"));

System.***out***.println(System.*getProperty*("user.home"));

System.***out***.println(System.*getProperty*("java.home"));

System.***out***.println(System.*getProperty*("os.name"));

}

}

**Lab Exercise No:**32

**Exercise Objective(s):***Scanner class*

**Exercise:***Create a class called Student. Get the details like name, degree, age, total marks and*

*percentage from the user and display the same.*

Solution 32:

**package** com.day.four;

**import** java.util.Scanner;

**public** **class** Student {

String name;

String degree;

**int** age;

**float** totalmarks;

**float** percentage;

@Override

**public** String toString() {

**return** "Student [name=" + name + ", degree=" + degree + ", age=" + age + ", totalmarks=" + totalmarks

+ ", percentage=" + percentage + "]";

}

**public** Student(String name, String degree, **int** age, **float** totalmarks, **float** percentage) {

**super**();

**this**.name = name;

**this**.degree = degree;

**this**.age = age;

**this**.totalmarks = totalmarks;

**this**.percentage = percentage;

}

**public** Student() {

**super**();

// **TODO** Auto-generated constructor stub

}

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

Student student=**new** Student();

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

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

student.name=sc.next();

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

student.degree=sc.next();

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

student.age=sc.nextInt();

System.***out***.println("Enter total marks");

student.totalmarks=sc.nextFloat();

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

student.percentage=sc.nextFloat();

student=**new** Student(student.name,student.degree,student.age,student.totalmarks,student.percentage);

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

sc.close();

}

}

**Lab Exercise No:**33

**Exercise Objective(s):***Systemclass,usingstaticimport*

**Exercise:***Create a Package called house. Create 2 classes namely Hall and Kitchen.*

1. *In the Hall class print the message “This is the first room while entering the house” without using the class name System explicitly in the println statement.*
2. *In the Kitchen class create an array called appliances and initialize with values and print the same.*
3. *After printing copy that array into a different array.*
4. *Invoke garbage collector explicitly for the Kitchen class.*

Solution 33:

//calling of static member of System class without Class name

**package** com.house;

**import** **static** java.lang.System.\*;

**public** **class** Hall {

//constructor to print the statement when object is created

Hall(){

***out***.println("“This is the first room while entering the house” ");

}

}

**package** com.house;

//calling of static member of System class without Class name

**import** **static** java.lang.System.\*;

**public** **class** Kitchen {

**void** KitchenAppliances() {

//array creation

String [] appliances= **new** String[]{"utensils","Gas","Fridge","lighter"};

//displaying array items

**for**(**int** index=0;index<appliances.length;index++){

***out***.println(appliances[index]);

}

String copyAppliances[]=appliances;//copying the appliances array in copyAppliances

System.*gc*();//invoking garbage collector explicitly for the Kitchen class

}

}

**package** com.house;

**public** **class** TestHallKitchen {

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

//creating objets and calling the methods

Hall hall=**new** Hall();

Kitchen k=**new** Kitchen();

k.KitchenAppliances();

}

}

**Lab Exercise No:**50

**Exercise Objective(s):***syntax*

**Exercise:***In the Lab Exercise 14, change the code such that the numbers are taken as input from the*

*user. Handle the appropriate exceptions.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***InputMismatchException, Arithmetic Exception*

Solution 50:

**package** com.day.four;

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

**public** **class** Calculator {

**public** **void** add(**int** a,**int** b){

**int** c=a+b;

System.***out***.println("Sum is : "+c);

}

**public** **void** diff(**int** a,**int** b){

**int** c=a-b;

System.***out***.println("Diff is : "+c);

}

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

**int** c=a\*b;

System.***out***.println("Product is : "+c);

}

**public** **void** div(**int** a,**int** b){

**try**

{

**int** c=a/b;

//if any arithmetical error occurs the complier will go to catch block and wont execute further

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

}

**catch**(ArithmeticException e )

{

System.***out***.print("Arithmetic Exception at division");

}

}

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

Calculator ob=**new** Calculator();//creating object

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

System.***out***.println("Enter 1 for addition");

System.***out***.println("Enter 2 for substraction");

System.***out***.println("Enter 3 for multiplication");

System.***out***.println("Enter 4 for division");

//try block to catch any exception occurred

**try**

{

System.***out***.println("Enter your first number: ");

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

System.***out***.println("Enter your second number: ");

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

System.***out***.println("Enter the operation to be performed");

**char** ch=sc.next().charAt(0);

**switch**(ch)

{

**case** '1' : ob.add(a, b);

**break**;

**case** '2' : ob.diff(a, b);

**break**;

**case** '3' : ob.mul(a, b);

**break**;

**case** '4' : ob.div(a, b);

**break**;

**default**: System.***out***.println("Entered wrong option");

}

}

**catch** (InputMismatchException e)

{

// if any wrong data type has been given as input the this exception will be executed

System.***out***.println("Input Mismatch Exception occured");

}

sc.close();

}

}

**Lab Exercise No:**51

**Exercise Objective(s):***syntax*

**Exercise:***In the Lab Exercise 17, handle the scenarios if the String variable is not initialized.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Null Pointer Exception*

Solution 51:

**package** com.day.four;

**public** **class** Solution51 {

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

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

**try**{

String s="The quick brown fox jumps over the lazy dog";

**char** c=s.charAt(11);

System.***out***.println("Character at the 12th index is : "+c);

System.***out***.println("the String contains the word “is” : " + s.contains("is"));

String s2=" and killed it";

String s3=s+s2;

System.***out***.println("After Appending"+s3);

**boolean** flag= s3.endsWith("dogs");

System.***out***.println("String ends with dog:"+flag);

// Checking whether the String is equal to “The quick brown Fox jumps over the lazy Dog”

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

**if**(s.equals("The quick brown Fox jumps over the lazy Dog"))

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

**else**

System.***out***.println("Not Equal");

//Checking whether the String is equal to “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”

**if**(s.compareTo("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG")==0)

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

**else**

System.***out***.println("Not Equal");

//Finding the index position of the character a

System.***out***.println(s.indexOf("a"));

//Finding the last index position of the character “e”

System.***out***.println(s.lastIndexOf("e"));

//Finding the length of the String.

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

//Checking whether the String matches to “The quick brown Fox jumps over the lazy Dog”

**if**(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

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

**else**

System.***out***.println("Not Equal");

//Replacing the word “The” with the word “A”

String s4=s.replace("The","A");

System.***out***.println("After Replacing the word “The” with the word “A”:"+s4);

//Splitting the above string into two such that two animal names do not come together

String arr[]=s.split(" ",s.length());

System.***out***.println("After splitting");

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

{

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

}

//Printing the animal names alone separately from the above string.

String t="",z="";

**for** (String val: s.split(" ")) {

**if**(val.equals("fox")){

t=val;

}

**if**(val.equals("dog")){

z=val;

}

}

// printing the final value.

System.***out***.println("Animal names are:"+t+" and "+z);

//Printing the above string in completely lower case

System.***out***.println("String in lowercase: "+s.toLowerCase());

//Printing the above string in completely upper case

System.***out***.println("String in uppercase: "+s.toUpperCase());

}

**catch**(NullPointerException e) //Null pointer Exception

{

e.printStackTrace();

}

}

}

**Lab Exercise No:**52

**Exercise Objective(s):***syntax*

**Exercise:***Using Lab Exercise 17, catch and demonstratethe required exceptions.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***StringIndexOutOfBoundsException*

Solution 52:

**package** com.day.four;

**public** **class** Solution52 {

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

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

**try**{

String s="The quick brown fox jumps over the lazy dog";

**char** c=s.charAt(11);

System.***out***.println("Character at the 12th index is : "+c);

System.***out***.println("the String contains the word “is” : " + s.contains("is"));

String s2=" and killed it";

String s3=s+s2;

System.***out***.println("After Appending"+s3);

**boolean** flag= s3.endsWith("dogs");

System.***out***.println("String ends with dog:"+flag);

// Checking whether the String is equal to “The quick brown Fox jumps over the lazy Dog”

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

**if**(s.equals("The quick brown Fox jumps over the lazy Dog"))

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

**else**

System.***out***.println("Not Equal");

//Checking whether the String is equal to “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”

**if**(s.compareTo("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG")==0)

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

**else**

System.***out***.println("Not Equal");

//Finding the index position of the character a

System.***out***.println(s.indexOf("a"));

//Finding the last index position of the character “e”

System.***out***.println(s.lastIndexOf("e"));

//Finding the length of the String.

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

//Checking whether the String matches to “The quick brown Fox jumps over the lazy Dog”

**if**(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

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

**else**

System.***out***.println("Not Equal");

//Replacing the word “The” with the word “A”

String s4=s.replace("The","A");

System.***out***.println("After Replacing the word “The” with the word “A”:"+s4);

//Splitting the above string into two such that two animal names do not come together

String arr[]=s.split(" ",s.length());

System.***out***.println("After splitting");

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

{

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

}

//Printing the animal names alone separately from the above string.

String t="",z="";

**for** (String val: s.split(" ")) {

**if**(val.equals("fox")){

t=val;

}

**if**(val.equals("dog")){

z=val;

}

}

// printing the final value.

System.***out***.println("Animal names are:"+t+" and "+z);

//Printing the above string in completely lower case

System.***out***.println("String in lowercase: "+s.toLowerCase());

//Printing the above string in completely upper case

System.***out***.println("String in uppercase: "+s.toUpperCase());

}

**catch**(StringIndexOutOfBoundsException e) //Exception if string index out of bound occurs

{

e.printStackTrace();

}

**catch**(Exception e) //General Exception

{

e.printStackTrace();

}

}

}

**Lab Exercise No:**53

**Exercise Objective(s):***syntax*

**Exercise:***Using Lab Exercise 22, catch and demonstrate the required exceptions.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***ArrayIndexOutOfBoundsException*

Solution 53:

**package** com.day.four;

**import** java.util.Scanner;

**public** **class** Solution53 {

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

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

**try**{

**int** row, column, c, d;

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

System.***out***.println("Enter the number of rows and columns of matrix");

row = sc.nextInt();

column = sc.nextInt();

**int** first[][] = **new** **int**[row][column]; //creating matrix

**int** square[][] = **new** **int**[row][column];

System.***out***.println("Enter the elements of first matrix"); //input of first 2D array

**for** (c = 0; c < row; c++)

**for** (d = 0; d < column; d++)

first[c][d] = sc.nextInt();

**for** (c = 0; c < row; c++)

**for** (d = 0; d < column; d++)

square[c][d] = first[c][d] \* first[c][d]; //squaring each element of the 2D array

System.***out***.println("Square of the matrices:");

//Displaying the matrix

**for** (c = 0; c < row; c++)

{

**for** (d = 0; d <= column; d++)//This statement will generate array index out of bound exception

System.***out***.print(square[c][d] + "\t");

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

}

}

**catch**(ArrayIndexOutOfBoundsException e)

{

e.printStackTrace();

}

}

}

**Lab Exercise No:**54

**Exercise Objective(s):***Exception class methods*

**Exercise:***By using multiple catch blocks, write a class to demonstrate the order of the execution of the*

*catch blocks usingNegativeArraySizeException,ArrayIndexOutOfBoundsException,*

*StringIndexOutOfBoundsException, IndexOutOfBoundsException, NullPointerException,*

*ArithmeticException and print the stack trace for each exception.*

Solution 54:

**package** com.day.four;

**import** java.util.Scanner;

**public** **class** ExceptionHandling54 {

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

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

**try**{

**int** num1=30;

System.***out***.println("Enter 2nd num");

**int** num2=sc.nextInt();

**int** output=num1/num2;

System.***out***.println ("Result: "+output);

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

//Array has only 5 elements

System.***out***.println("Enter the index to be accessed and array size is 5");

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

System.***out***.println("Element is: "+a[index]);

System.***out***.println("Enter the size of array which is not negative other will throw exception");

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

**int**[] myArray = **new** **int**[arrSize];

String str="beginnersbook";

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

**char** c = str.charAt(0);

System.***out***.println("Enter the index to be accessed from string");

**int** index2=sc.nextInt();

c = str.charAt(index2);

System.***out***.println("character is : "+c);

String str2 = "Java Code Geeks";

System.***out***.println("Enter the index to get the substring");

**int** index3=sc.nextInt();

//The following statement throws an exception, because the request index is invalid.

String subStr = str2.substring(index3);

System.***out***.println("Substring is: "+subStr);

}

**catch**(ArithmeticException e){

System.***out***.println ("You Shouldn't divide a number by zero");

e.printStackTrace();

}

**catch**(ArrayIndexOutOfBoundsException e){

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

e.printStackTrace();

}

**catch**(StringIndexOutOfBoundsException e){

System.***out***.println("StringIndexOutOfBoundsException!!");

e.printStackTrace();

}

**catch**(NullPointerException e){

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

e.printStackTrace();

}

**catch** (NegativeArraySizeException ex) {

System.***out***.println("Can't create array of negative size");

ex.printStackTrace();

}

}

}

**Lab Exercise No:**55

**Exercise Objective(s):***User-defined exceptions*

**Exercise:***In the Lab Exercise 46, handle the expected exceptions by writing custom defined exceptions.*

*Solution 55:*

*package com.bank;*

*public interface Account {*

*String savings="Savings";*

*String fixed="fixed";*

*String personalLoan="personal Loan";*

*String housingLoan="housing Loan";*

*void createAcc();*

*}*

*package com.bank;*

*public interface CreditInterest extends Interest{*

*double addMonthlyInt ();*

*double addHalfYrlyInt();*

*double addAnnualInt();*

*}*

*package com.bank;*

*public interface DebitInterest extends Interest{*

*double deductMonthlyInt();*

*double deductHalfYrlyInt ();*

*double deductAnnualInt ();*

*}*

*package com.bank;*

*public interface DepositAcc extends Account{*

*double withdraw (double amt);*

*double deposit(double da);*

*double getBalance();*

*}*

*package com.bank;*

*public interface Interest {*

*double savingsInterest=6;*

*double fixedInterest=8.5;*

*double personalLoanInterest=10;*

*double housingLoanInterest=11;*

*double calcInt();*

*}*

*package com.bank;*

*public interface LoanAcc extends Account{*

*double repayPrincipal (double principal);*

*double payInterest (double interest);*

*double payPartialPrincipal (double partialPrincipal);*

*}*

**package** com.bank;

**import** com.BankImpl.\*;

**public** **class** MyAccount{

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

SavingsAcc sc=**new** SavingsAcc();

sc.createAcc();

sc.setBalance(1000);

sc.calcInt();

sc.addAnnualInt();

sc.addHalfYrlyInt();

sc.addMonthlyInt();

sc.deposit(800);

sc.withdraw(20000);

sc.deposit(1000);

System.***out***.println("Balance is: "+sc.getBalance());

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

FDAcc fd=**new** FDAcc();

fd.createAcc();

fd.setBalance(1000);

fd.calcInt();

fd.addAnnualInt();

fd.addHalfYrlyInt();

fd.addMonthlyInt();

fd.deposit(800);

fd.withdraw(200);

System.***out***.println("Balance is: "+fd.getBalance());

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

HousingLoanAcc hla=**new** HousingLoanAcc();

hla.createAcc();

hla.setBalance(5000);

hla.calcInt();

hla.deductMonthlyInt();

hla.deductHalfYrlyInt();

hla.deductAnnualInt();

hla.repayPrincipal(400);

hla.payInterest(200);

hla.payPartialPrincipal(300);

System.***out***.println("Balance is: "+hla.getBalance());

PersonalLoanAcc pla=**new** PersonalLoanAcc();

pla.createAcc();

pla.setBalance(10000);

pla.calcInt();

pla.deductMonthlyInt();

pla.deductHalfYrlyInt();

pla.deductAnnualInt();

pla.repayPrincipal(600);

pla.payInterest(200);

pla.payPartialPrincipal(300);

System.***out***.println("Balance is: "+pla.getBalance());

}

}

package com.BankImpl;

import com.bank.CreditInterest;

import com.bank.DepositAcc;

import com.bank.MyException;

public class FDAcc implements DepositAcc, CreditInterest{

private double balance;

public void setBalance(double balance) {

this.balance = balance;

}

@Override

public void createAcc() {

// TODO Auto-generated method stub

System.out.println(fixed+" account created");

}

@Override

public double calcInt() {

// TODO Auto-generated method stub

System.out.println(savings+" account interest is:"+fixedInterest);

double interest=(this.balance\*fixedInterest)/100;

this.balance=this.balance+interest;

return interest;

}

@Override

public double addMonthlyInt() {

// TODO Auto-generated method stub

System.out.println(fixed+" account monthly interest is:"+fixedInterest/12);

double interest=(this.balance\*fixedInterest)/100;

this.balance=this.balance+interest;

return interest;

}

@Override

public double addHalfYrlyInt() {

// TODO Auto-generated method stub

System.out.println(fixed+" half yearly interest is :"+fixedInterest/2);

this.balance=this.balance+(this.balance\*fixedInterest)/100;

return this.balance;

}

@Override

public double addAnnualInt() {

// TODO Auto-generated method stub

System.out.println(fixed+" addAnnualInt"+fixedInterest);

this.balance=this.balance+(this.balance\*fixedInterest)/100;

return this.balance;

}

@Override

public double withdraw(double w) {

// TODO Auto-generated method stub

try {

if(this.balance>w){

this.balance= (this.balance-w);

}

else{

throw new MyException();

}

}

catch (MyException me) {

me.withdrawalException();

}

return this.balance;

}

@Override

public double deposit(double da) {

// TODO Auto-generated method stub

this.balance=(this.balance+da);

return (this.balance);

}

@Override

public double getBalance() {

return this.balance;

}

}

*package com.BankImpl;*

*import com.bank.DebitInterest;*

*import com.bank.LoanAcc;*

*public class HousingLoanAcc implements LoanAcc , DebitInterest{*

*private double balance;*

*public void setBalance(double balance) {*

*this.balance = balance;*

*}*

*@Override*

*public void createAcc() {*

*// TODO Auto-generated method stub*

*System.out.println(housingLoan+" account created");*

*}*

*@Override*

*public double calcInt() {*

*// TODO Auto-generated method stub*

*System.out.println(housingLoan+" account interest is:"+housingLoanInterest);*

*double interest=(this.balance\*housingLoanInterest)/100;*

*this.balance=this.balance+interest;*

*return interest;*

*}*

*@Override*

*public double deductMonthlyInt() {*

*// TODO Auto-generated method stub*

*System.out.println(housingLoan+" account monthly interest is:"+housingLoanInterest/12);*

*double interest=(this.balance\*housingLoanInterest)/100;*

*this.balance=this.balance-interest;*

*return interest;*

*}*

*@Override*

*public double deductHalfYrlyInt() {*

*// TODO Auto-generated method stub*

*System.out.println(housingLoan+" account deductHalfYrlyInt is:"+housingLoanInterest/2);*

*double interest=(this.balance\*housingLoanInterest)/100;*

*this.balance=this.balance-interest;*

*return interest;*

*}*

*@Override*

*public double deductAnnualInt() {*

*// TODO Auto-generated method stub*

*System.out.println(housingLoan+" account deductAnnualInt is:"+housingLoanInterest);*

*double interest=(this.balance\*housingLoanInterest)/100;*

*this.balance=this.balance-interest;*

*return interest;*

*}*

*@Override*

*public double repayPrincipal(double p) {*

*// TODO Auto-generated method stub*

*this.balance-=p;*

*return this.balance;*

*}*

*@Override*

*public double payInterest(double p) {*

*// TODO Auto-generated method stub*

*this.balance-=p;*

*return this.balance;*

*}*

*@Override*

*public double payPartialPrincipal(double p) {*

*// TODO Auto-generated method stub*

*this.balance-=p;*

*return this.balance;*

*}*

*public double getBalance() {*

*return this.balance;*

*}*

*}*

*package com.BankImpl;*

*import com.bank.DebitInterest;*

*import com.bank.LoanAcc;*

*public class PersonalLoanAcc implements LoanAcc, DebitInterest{*

*private double balance;*

*public void setBalance(double balance) {*

*this.balance = balance;*

*}*

*@Override*

*public void createAcc() {*

*// TODO Auto-generated method stub*

*System.out.println(personalLoan+" account created");*

*}*

*@Override*

*public double calcInt() {*

*// TODO Auto-generated method stub*

*System.out.println(personalLoan+" account interest is:"+personalLoanInterest);*

*double interest=(this.balance\*personalLoanInterest)/100;*

*this.balance=this.balance+interest;*

*return interest;*

*}*

*@Override*

*public double deductMonthlyInt() {*

*// TODO Auto-generated method stub*

*System.out.println(personalLoan+" account monthly interest is:"+personalLoanInterest/12);*

*double interest=(this.balance\*personalLoanInterest)/100;*

*this.balance=this.balance-interest;*

*return interest;*

*}*

*@Override*

*public double deductHalfYrlyInt() {*

*// TODO Auto-generated method stub*

*System.out.println(personalLoan+" account deductHalfYrlyInt is:"+personalLoanInterest/2);*

*double interest=(this.balance\*personalLoanInterest)/100;*

*this.balance=this.balance-interest;*

*return interest;*

*}*

*@Override*

*public double deductAnnualInt() {*

*// TODO Auto-generated method stub*

*System.out.println(personalLoan+" account deductAnnualInt is:"+personalLoanInterest);*

*double interest=(this.balance\*personalLoanInterest)/100;*

*this.balance=this.balance-interest;*

*return interest;*

*}*

*@Override*

*public double repayPrincipal(double p) {*

*// TODO Auto-generated method stub*

*this.balance-=p;*

*return this.balance;*

*}*

*@Override*

*public double payInterest(double p) {*

*// TODO Auto-generated method stub*

*this.balance-=p;*

*return this.balance;*

*}*

*@Override*

*public double payPartialPrincipal(double p) {*

*// TODO Auto-generated method stub*

*this.balance-=p;*

*return this.balance;*

*}*

*public double getBalance() {*

*return this.balance;*

*}*

*}*

*package com.BankImpl;*

*import com.bank.Account;*

*import com.bank.CreditInterest;*

*import com.bank.DepositAcc;*

*import com.bank.MyAccount;*

*import com.bank.MyException;*

*public class SavingsAcc implements DepositAcc,CreditInterest{*

*private double balance;*

*public void setBalance(double balance) {*

*this.balance = balance;*

*}*

*@Override*

*public void createAcc() {*

*// TODO Auto-generated method stub*

*System.out.println(savings+" account created");*

*}*

*@Override*

*public double calcInt() {*

*// TODO Auto-generated method stub*

*System.out.println(savings+" account interest is:"+savingsInterest);*

*double interest=(this.balance\*savingsInterest)/100;*

*this.balance=this.balance+interest;*

*return interest;*

*}*

*@Override*

*public double addMonthlyInt() {*

*// TODO Auto-generated method stub*

*System.out.println(savings+" account monthly interest is:"+savingsInterest/12);*

*double interest=(this.balance\*savingsInterest)/100;*

*this.balance=this.balance+interest;*

*return interest;*

*}*

*@Override*

*public double addHalfYrlyInt() {*

*// TODO Auto-generated method stub*

*System.out.println(savings+" half yearly interest is :"+savingsInterest/2);*

*this.balance=this.balance+(this.balance\*savingsInterest)/100;*

*return this.balance;*

*}*

*@Override*

*public double addAnnualInt() {*

*// TODO Auto-generated method stub*

*System.out.println(savings+" addAnnualInt "+savingsInterest);*

*this.balance=this.balance+(this.balance\*savingsInterest)/100;*

*return this.balance;*

*}*

*@Override*

*public double withdraw(double w) {*

*// TODO Auto-generated method stub*

*try {*

*if(this.balance>w){*

*this.balance= (this.balance-w);*

*}*

*else{*

*throw new MyException();*

*}*

*}*

*catch (MyException me) {*

*me.withdrawalException();*

*}*

*return this.balance;*

*}*

*@Override*

*public double deposit(double da) {*

*// TODO Auto-generated method stub*

*this.balance=(this.balance+da);*

*return (this.balance);*

*}*

*@Override*

*public double getBalance() {*

*return this.balance;*

*}*

*}*

**Lab Exercise No:**56

**Exercise Objective(s):***finally keyword*

**Exercise:***Create a class such that it resets the value of the objects it used to null after its usage in all*

*cases.*

Solution 56:

**package** com.day.four;

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

**public** **class** Solution56 {

**int** a;

**int** b;

**public** Solution56() {

**super**();

}

**public** Solution56(**int** a **int** b) {

**super**();

**this**.a = a;

**this**.b = b;

}

@Override

**public** String toString() {

**return** "Solution56 [num1=" + a + ", num2=" + b + "]";

}

**public** **void** add(**int** a,**int** b){

**int** c=a+b;

System.***out***.println("Sum is : "+c);

}

**public** **void** diff(**int** a,**int** b){

**int** c=a-b;

System.***out***.println("Diff is : "+c);

}

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

**int** c=a\*b;

System.***out***.println("Product is : "+c);

}

**public** **void** div(**int** a,**int** b){

**try**

{

**int** c=a/b;

//if any arithmetical error occurs the complier will go to catch block and wont execute further

System.***out***.println("quotient is: "+c);

}

**catch**(ArithmeticException e )

{

System.***out***.print("Arithmetic Exception at division");

}

}

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

Solution56 ob=**new** Solution56();//creating object

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

System.***out***.println("Enter 1 for addition");

System.***out***.println("Enter 2 for substraction");

System.***out***.println("Enter 3 for multiplication");

System.***out***.println("Enter 4 for division");

//try block to catch any exception occurred

**try**

{

System.***out***.println("Enter your first number: ");

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

System.***out***.println("Enter your second number: ");

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

System.***out***.println("Enter the operation to be performed");

**char** ch=sc.next().charAt(0);

**switch**(ch)

{

**case** '1' : ob.add(x,y);

**break**;

**case** '2' : ob.diff(x,y);

**break**;

**case** '3' : ob.mul(x,y);

**break**;

**case** '4' : ob.div(x,y);

**break**;

**default**: System.***out***.println("Entered wrong option");

}

}

**catch** (InputMismatchException e)

{

// if any wrong data type has been given as input the this exception will be executed

System.***out***.println("Input Mismatch Exception occured");

}

**finally**{

System.***out***.println("Setting object to null after use");

ob=**null**;

ob.toString();

}

sc.close();

}

}

**Lab Exercise No:**57

**Exercise Objective(s):***finally keyword*

**Exercise:***Create a class such that a method uses the try catch block with the return type of String.*

Solution 57:

**package** com.day.four;

**public** **class** Solution57 {

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

{

//calling a static method

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

}

**static** String methodReturningValue()

{

String s = **null**;

**try**

{

s = "return value from try block";

**return** s;

//Will return string from try block and wont execute any further

}

**catch** (Exception e)

{

s = s + "return value from catch block";

**return** s;

}

**finally**

{

s = s + "return value from finally block";

}

}

}

**Lab Exercise No:**58

**Exercise Objective(s):***User-defined exceptions*

**Exercise:***Createa class called Employee which asks the user to input the name and the age of a*

*employee. Raise a custom defined exception when the user enters an employee name*

*that has been already entered and raise another exception if the age is negative or less*

*than 18 or greater than 60.*

Solution 58:

**package** com.day.four;

**import** java.util.Scanner;

//Defining Our own exception by extending Exception class

**class** AgeException **extends** Exception {

**public** AgeException(String str) {

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

}

}

//Defining Our own exception by extending Exception class

**class** NameException **extends** Exception{

**public** NameException(String st){

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

}

}

**public** **class** UserDefinedException58 {

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

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

**try** {

String employee[]=**new** String[]{"aa","ccc","dddd","qqq"};

System.***out***.println("Enter employee name");

String emp=s.next();

//checking for employee already present

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

**if**(emp.equals(employee[i])){

//throwing to user defined exception class

**throw** **new** NameException("Duplicate names");

}

}

System.***out***.println("unique employee ");

System.***out***.print("Enter ur age : ");

**int** age = s.nextInt();

**if**(age < 18 || age> 60)

//checking age invalid or not

**throw** **new** AgeException("Invalid age");

**else**

System.***out***.println("Valid age");

}

**catch** (AgeException a) {

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

}

**catch** (NameException e){

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

}

s.close();

}

}