**JAVA**

**Packages**

**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:**

/\* Understanding the concept of package and calling of different class from the class the within the package \*/

package shapes;

public class Circle {

void circleDisplay() { //display function of circle class

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

}

}

public class Square {

void squareDisplay() { //display function of square class

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

}

}

public class Triangle {

void triangleDisplay() { //display function of triangle class

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

}

}

public class TestShapes { //TestShapes calls Circle,Square and Triangle class within the same package

public static void main(String[] args) {

// TODO Auto-generated method stub

Circle c=new Circle();

c.circleDisplay();

Square s=new Square();

s.squareDisplay();

Triangle t=new Triangle();

t.triangleDisplay();

}

}

**--------------------------------------------------------------------------------------------------------------------------------------**

**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.*

**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:**

/\* This is a java program to display JVM information.\*/

package com.hsbc.practice;

import java.util.Properties;

public class JvmProperty {

public static void main(String[] args) {

// TODO Auto-generated method stub

Properties p = System.getProperties();

//Listing the JVM properties

p.list(System.out);

}

}

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

**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:**

/\* To understand the use of Scanner class for taking input from the user \*/

package com.hsbc.practice;

import java.util.Scanner;

public class Student {

String name;

String degree;

int age;

float totalmarks;

float 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();

}

@Override

public String toString() {

return "Student has folloeing data\*\*\*\*\*name=" + name + ", degree=" + degree + ", age=" + age + ", totalmarks=" + totalmarks + ", percentage=" + percentage + "";

}

public static void main(String[] args) {

// TODO Auto-generated method stub

Student stud=new Student();

Scanner sc=new Scanner(System.in);

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

stud.name=sc.next();

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

stud.degree=sc.next();

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

stud.age=sc.nextInt();

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

stud.totalmarks=sc.nextFloat();

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

stud.percentage=sc.nextFloat();

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

System.out.println(stud.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:**

/\* Java program to print a statement without the use of System class explicitly and calling of garbage collector when needed \*/

package house;

import java.io.FileDescriptor;

import java.io.FileOutputStream;

import java.io.PrintStream;

public class ThirtyThreeSol {

public static void main(String[] args) {

// TODO Auto-generated method stub

Hall h1 = new Hall();

System.out.println();

Kitchen k1 = new Kitchen();

}

}

class Hall{

Hall(){

PrintStream myout = new PrintStream(new FileOutputStream(FileDescriptor.out));

myout.print("This is the first room while entering the house");

}

}

class Kitchen{

String[] appliances = {"Fridge","Oven","Toaster","Cooker","Grinder"};

Kitchen(){

System.out.println("The appliances found the kitchen are as follows : ");

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

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

}

String[] copy = appliances;

//Invoking the garbage collector explicitly

System.gc();

}

}

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

**JAVA**

**Exception Handling**

**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.*

**SOLUTION:**

/\* Understanding the use of exception while performing basic arithmetic operation \*/

package com.hsbc.practice;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Calculator {

int num1,num2,res;

void add(int num1,int num2) //functions adds the two integer values and displays

{

res=num1+num2;

System.out.println("Result:" +num1 +"+"+num2+ "=" +res);

}

void sub(int num1,int num2) //functions subtracts the two integer values and displays

{

res=num1-num2;

System.out.println("Result:" +num1 +"-"+num2+ "=" +res);

}

void mult(int num1,int num2) //functions multiplies the two integer values and displays

{

res=num1\*num2;

System.out.println("Result:" +num1 +"\*"+num2+ "=" +res);

}

void div(int num1,int num2) //functions divides the two integer values and displays

{

try

{

res=num1/num2; //an arithmetic exception would be thrown if num2 is 0

}

catch(ArithmeticException e )

{

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

}

Finally //finally block will execute irrespective of exception thrown

{

System.out.println("Result:" +num1 +"/"+num2+ "=" +res);

}

}

public static void main(String[] args) {

int a,b;

char ch;

Calculator c=new Calculator();

Scanner sc=new Scanner(System.in);

try

{

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

a = sc.nextInt();

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

b = sc.nextInt();

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

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

switch(ch)

{

case '+' : c.add(a, b);

break;

case '-' : c.sub(a, b);

break;

case '\*' : c.mult(a, b);

break;

case '/' : c.div(a, b);

break;

default: System.out.println("Wrong operartion eneterd");

}

}

catch (InputMismatchException e) //this exception is thrown when the two values are not as per to the requirement

{

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

}

}

}

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

**Lab Exercise No:**51

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

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

**SOLUTION:**

**//** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Handling the String exceptions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package com.hsbc.practice;

public class Syntax {

public static void main(String[] args) {

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

try {

System.out.println("The character at 12th place is " + s.charAt(121));

}

catch(Exception e) {

e.printStackTrace();

System.out.println("This is called String Index Out of Bounds Exception. You are trying to access content of a String at a position whose location is not allotted.");

}

try {

System.out.println("'is' exists in the string 's' : " + s.contains("is"));

s = s + " and killed it";

System.out.println("The new string is :" + s);

System.out.println("The string ends with 'dogs' :" + s.endsWith("dogs"));

String s1 = "The quick brown Fox jumps over the lazy Dog";

if(s1.equals(s))

System.out.println("The strings are equal");

else

System.out.println("The strings are not equal");

String s2 = "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG";

if(s2.equals(s))

System.out.println("The strings are equal");

else

System.out.println("The strings are not equal");

System.out.println("The index of first occurence of 'a' is: " + s.indexOf("a"));

System.out.println("The index of last occurence of 'e' is: " + s.lastIndexOf("e"));

System.out.println("Length of string 's' is: " + s.length());

String s3 = "The quick brown Fox jumps over the lazy Dog";

if(s3.equals(s))

System.out.println("The strings are equal");

else

System.out.println("The strings are not equal");

s = s.replaceAll("The","A");

System.out.println("The new string is: " + s);

String sp1 = s.substring(0, 28);

String sp2 = s.substring(28);

System.out.println(sp1 + " | " + sp2);

String[] sl = s.split(" ");

System.out.println("The animal names are : " + sl[3] + " and " + sl[8]);

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

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

System.out.println("To check for null pointer exception.");

s=null;

System.out.println(s.charAt(8));

}

catch(NullPointerException e) {

e.printStackTrace();

System.out.println("Trying to access content of a String that has been already declared as null");

}

}

}

*-----------------*------------------------------------------------------------------------------------------------------------------

**Lab Exercise No:**52

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

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

**SOLUTION:**

**//** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Handling the String exceptions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package com.hsbc.practice;

public class Syntax {

public static void main(String[] args) {

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

try {

System.out.println("The character at 12th place is " + s.charAt(121));

}

catch(Exception e) {

e.printStackTrace();

System.out.println("This is called String Index Out of Bounds Exception. You are trying to access content of a String at a position whose location is not allotted.");

}

try {

System.out.println("'is' exists in the string 's' : " + s.contains("is"));

s = s + " and killed it";

System.out.println("The new string is :" + s);

System.out.println("The string ends with 'dogs' :" + s.endsWith("dogs"));

String s1 = "The quick brown Fox jumps over the lazy Dog";

if(s1.equals(s))

System.out.println("The strings are equal");

else

System.out.println("The strings are not equal");

String s2 = "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG";

if(s2.equals(s))

System.out.println("The strings are equal");

else

System.out.println("The strings are not equal");

System.out.println("The index of first occurence of 'a' is: " + s.indexOf("a"));

System.out.println("The index of last occurence of 'e' is: " + s.lastIndexOf("e"));

System.out.println("Length of string 's' is: " + s.length());

String s3 = "The quick brown Fox jumps over the lazy Dog";

if(s3.equals(s))

System.out.println("The strings are equal");

else

System.out.println("The strings are not equal");

s = s.replaceAll("The","A");

System.out.println("The new string is: " + s);

String sp1 = s.substring(0, 28);

String sp2 = s.substring(28);

System.out.println(sp1 + " | " + sp2);

String[] sl = s.split(" ");

System.out.println("The animal names are : " + sl[3] + " and " + sl[8]);

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

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

System.out.println("To check for null pointer exception.");

s=null;

System.out.println(s.charAt(8));

}

catch(NullPointerException e) {

e.printStackTrace();

System.out.println("Trying to access content of a String that has been already declared as null");

}

}

}

*------*--------------------------------------------------------------------------------------------------------------------------------

**Lab Exercise No:**53

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

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

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*understanding 2D array and operations performed over them\*\*\*\*\*\*\*\*\*\*

public class SquareMatrix

{

  public static void main(String args[])

  {

    int row, column, c, d;

    row = 3;

    column = 3;

    int first[][] = {{1,2,3},{4,5,6},{7,8,9}}      //creating and initializing matrix

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

    try

{

    for (c = 0; c < row; c++)

       for (d = 0; d < column; d++)

         square[c][d] = first[c][d] \* first[c][d];   //squaring each element through loop

}

Catch(ArrayIndexOutOfBoundsException e)

{

e.printStackTrace();

}

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

    for (c = 0; c < row; c++)

    {

       for (d = 0; d < column; d++)

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

      System.out.println();

    }

  }

}

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

**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:**

package com.hsbc.practice;

public class Solution54 {

void StringExceptions(String s) //demontrate different exceptions of String class

{

try

{

char a=s.charAt(20);

s=null;

System.out.println(s);

}

catch(StringIndexOutOfBoundsException e)

{

e.printStackTrace();

System.out.println("Tried accessing String element out of range");

}

catch(NullPointerException e)

{

e.printStackTrace();

System.out.println("The string is null");

}

}

void ArrayExceptions(int a[]) //demontrate different exceptions while working with array

{

try

{

int n=a.length;

System.out.println(a[n+1]);

a=new int[-1];

System.out.println(a);

}

catch(ArrayIndexOutOfBoundsException e)

{

e.printStackTrace();

System.out.println("Tried accessing array element out of range");

}

catch(NegativeArraySizeException e)

{

e.printStackTrace();

System.out.println("array has negative size");

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

String s="Demo String";

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

Solution54 s1=new Solution54();

s1.ArrayExceptions(a);

s1.StringExceptions(s);

}

}

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

**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.*

**Recommended duration:** *20 Mins*

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

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

**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:**

/\* understanding the use of finally keyword and creating a class which resets its object to null once it complete all the tasks \*/

package com.hsbc.practice;

public class Solution56 {

int num1;

int num2;

public Solution56() {

super();

}

public Solution56(int num1, int num2) {

super();

this.num1 = num1;

this.num2 = num2;

}

void div(int num1,int num2)

{

int res;

try

{

res=num1/num2;

System.out.println("Result:" +res);

}

catch(ArithmeticException e)

{

System.out.println("Arithmetic Exception at divison");

}

}

void add(int num1,int num2)

{

int res;

res=num1+num2;

System.out.println("Result:" +res);

}

void sub(int num1,int num2)

{

int res;

res=num1-num2;

System.out.println("Result:" +res);

}

void mult(int num1,int num2)

{

int res;

res=num1\*num2;

System.out.println("Result:" +res);

}

public static void main(String args[]) { /\* after all the arithmetic functions are performed, s

will be set to null\*/

Solution56 s=new Solution56();

try

{

s.add(100,10);

s.sub(100,10);

s.mult(100,10);

s.div(100,10);

}

catch(ArithmeticException e)

{

System.out.println("Exception while calling");

}

Finally

{

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

s=null;

System.out.println(s);

}

}

}

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

**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:**

//\*\*\*\*\*\*\*\*\* Understanding the use of try-catch block with the return type String\*\*\*\*\*\*\*\*\*

package com.hsbc.practice;

public class FinallyDemo {

public String test(String s) {

try {

if (s==null)

throw new Exception();

return s;

}

catch (Exception e) {

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

}

finally {

System.out.println("Printing in finally block");

}

return s;

}

public static void main(String[] args) {

FinallyDemo f=new FinallyDemo();

System.out.println(f.test("Aiman"));

System.out.println(f.test("Sahil"));

}

}

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

**Lab Exercise No:**58

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

**Exercise:** *Create a 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:**

//understanding the custom exceptions with the array of objects with the help ArrayList collection

package com.hsbc.practice;

import java.util.ArrayList;

class CustomException extends Exception{ /\* displaying two exceptions one with age and the other

with name \*/

public void display1() //age limit:- 18-60

{

System.out.println("Age should be in the range of 18-60");

}

public void display2() //unique names are allowed

{

System.out.println("This username is not available. Please enter another");

}

}

public class Employee {

/\*\*

\* @param args

\* @param eName

\* @param eAge

\*/

int eAge;

String eName;

private static ArrayList <String> al=new ArrayList<String>();

public Employee(int eAge, String eName) {

//super();

if((eAge>18 && eAge<60)&&(!al.contains(eName))) { /\* checking if the array list

already contains the name entered \*/

this.eAge = eAge;

this.eName = eName;

al.add(eName);

System.out.println(eAge + "\t" + eName);

}

else{

if((eAge<18||eAge>60))

try {

throw new CustomException1();

}

catch(CustomException1 ex){

ex.display1();

}

if(al.contains(eName)) {

try {

throw new CustomException1();

}

catch(CustomException1 ex){

ex.display2();

}}

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

//Employee58 e1 =

// ArrayList<String> al = new ArrayList<String>();

new Employee(12,"Aiman");

// e1.display();

//Employee58 e2 =

new Employee(-12,"Aiman");

new Employee(65,"Supriya");

//e3.display();

//Employee58 e4 =

new Employee(32,"Aditya");

//e4.display();

}

}

-------------------------------------------------------------END------------------------------------------------------------------