**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.*

**Recommended duration:** *20 Mins*

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

**package** com.hsbc.shapes;

/\*

\* Making a Shapes package and then defining some classes in it.

\* Displaying general message of each class.

\* \*/

**class** Square{

**public** **void** display(){

System.***out***.println(" Object of class Square ");

}

}

**class** Triangle{

**public** **void** display() {

System.***out***.println(" Object of class Triangle ");

}

}

**class** Circle{

**public** **void** display() {

System.***out***.println(" Object of class Circle ");

}

}

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

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

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

Triangle t = **new** Triangle();

Circle c = **new** Circle();

s.display();

t.display();

c.display();

}

}

**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.*

**File-1)**

**package** org.animal;

/\*

\* Lion Class

\* \*/

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

String color;

**int** weight,age;

//parameterized constructor

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

**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 can't climb trees.");

}

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

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

}

}

**File-2)**

**package** org.animal;

/\*

\* Tiger Class

\* \*/

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

String color;

**int** weight, age;

//parameterized constructor

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

**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 vegetarian.");

}

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

System.***out***.println("Tiger can climb trees.");

}

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

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

}

}

File-3)

**package** org.animal;

/\*

\* Deer Class

\* \*/

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

String color;

**int** weight, age;

//parameterized constructor

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

**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 vegetarians.");

}

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

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

}

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

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

}

}

**Main File)**

**package** org.animal;

// Usind all Animal class after exporting the jar file.

**import** org.animal.Deer;

**import** org.animal.Lion;

**import** org.animal.Tiger;

**class** VandalurZoo{

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

Lion lion =**new** Lion("Lion1",150,15);

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

lion.getColor();

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

lion.getAge();

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

lion.getWeight();

System.***out***.print("Can climb? : ");

lion.canClimb();

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

lion.isVegetarian();

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

lion.sound();

Tiger tiger =**new** Tiger("Tiger1",140,16);

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

tiger.getColor();

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

tiger.getAge();

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

tiger.getWeight();

System.***out***.print("Can climb? : ");

tiger.canClimb();

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

tiger.isVegetarian();

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

tiger.sound();

Deer deer =**new** Deer("Deer1",50,12);

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

deer.getColor();

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

deer.getAge();

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

deer.getWeight();

System.***out***.print("Can 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*

**Recommended duration:***10Mins*

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

**package** com.hsbc.lab4;

/\*

\* Program to display certain properties using System class.

\* \*/

**import** java.lang.System;;

**public** **class** Solution31 {

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

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

System.***out***.println("Java Version : "+System.*getProperty*("java.version"));

System.***out***.println("Java Vendor : "+System.*getProperty*("java.vendor"));

System.***out***.println("Class Path : "+System.*getProperty*("java.class.path"));

System.***out***.println("Installed Home directory : "+System.*getProperty*("user.home"));

System.***out***.println("OS name : "+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.*

**Recommended duration:** *20 Mins*

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

**package** com.hsbc.lab4;

/\*

\* Program to display use of Scanner class.

\* \*/

**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 information: name=" + name + ", degree=" + degree + ", age=" + age + ", totalmarks=" + totalmarks

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

}

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

Student s =**new** Student();

// instantiating of scanner class

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

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

// to get string

s.name=sc.next();

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

s.degree=sc.next();

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

// to get integer value

s.age=sc.nextInt();

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

// to get float value

s.totalmarks=sc.nextFloat();

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

s.percentage=sc.nextFloat();

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

System.***out***.println(s.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.*

**Recommended duration:***20Mins*

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

**package** com.hsbc.house;

/\*

\* Using System to make copy and Invoking garbage collector explicitly for Kitchen class.

\* \*/

**import** java.lang.System;

**class** Hall{

static{

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

}

}

**class** Kitchen{

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

Kitchen(){

System.***out***.println("The appliances in kitchen are: ");

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

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

}

String[] copy = **new** String[appliances.length];

System.*arraycopy*(appliances, 0, copy, 0, appliances.length);

System.*gc*();

}

}

**public** **class** House {

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

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

Hall h = **new** Hall();

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

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

}

}

**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.*

**Recommended duration:** *20 Mins*

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

**package** com.hsbc.lab4;

/\*

\* Handing exception explicitly

\* \*/

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

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

**int** num1,num2,res;

// Different functions of calculator

**void** add(**int** num1,**int** num2)

{

res=num1+num2;

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

}

**void** sub(**int** num1,**int** num2)

{

res=num1-num2;

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

}

**void** mult(**int** num1,**int** num2)

{

res=num1\*num2;

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

}

**void** div(**int** num1,**int** num2)

{

**try**

{

res=num1/num2;

}

**catch**(ArithmeticException e )

{

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

}

**finally**

{

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

}

}

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

**int** var1,var2;

**char** ch;

Calculator c=**new** Calculator();

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

**try**

{

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

var1 = sc.nextInt();

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

var2 = sc.nextInt();

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

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

**switch**(ch)

{

**case** '+' : c.add(var1,var2);

**break**;

**case** '-' : c.sub(var1,var2);

**break**;

**case** '\*' : c.mult(var1,var2);

**break**;

**case** '/' : c.div(var1,var2);

**break**;

**default**: System.***out***.println("Not valid");

}

}

**catch** (InputMismatchException e) // exception raised

{

System.***out***.println("Wrong input data type!!");

}

}

}

**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*

**package** com.hsbc.lab4;

/\*

\* Handling NullPointerException

\* \*/

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

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

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

String str11=**null**;

**try**{

**if**(str11==**null**){

**throw** **new** NullPointerException("Null value");

}}

**catch** (Exception e){

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

}

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

System.***out***.println("12th index character is : "+ str.charAt(12));

System.***out***.println("Word 'is' is present : "+ str.contains("is")); // 2.Check whether the String contains the word “is”.

str += " and killed it";

System.***out***.println("Appended string : "+ str); //3.Add the string “and killed it” to the existing string.

System.***out***.println("String ends with 'dogs\*' : "+ str.endsWith("dogs")); // 4.Check whether the String ends with the word “dogs”.

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

String checkingString2 = "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG";

System.***out***.println("String is equal with (The quick brown Fox jumps over the lazy Dog) : "+ (str==checkingString)); //5.Check whether the String is equal to “The quick brown Fox jumps over the lazy Dog”.

System.***out***.println("String is equal with (THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG) : "+ (str==checkingString2)); //6.Check whether the String is equal to “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”.

System.***out***.println("Index of char 'a' is : "+ str.indexOf("a")); //7.Find the index position of the character “a”.

System.***out***.println("Last index of char 'e' is : "+ str.lastIndexOf("e")); //8.Find the last index position of the character “e”.

System.***out***.println("Length of string is "+ str.length()); //9.Find the length of the String.

System.***out***.println("String is matches with (The quick brown Fox jumps over the lazy Dog) : "+ (str.matches(checkingString))); //10.Check whether the String matches to “The quick brown Fox jumps over the lazy Dog”.

System.***out***.println("After replacing word 'The' with 'A' : "+ str.replace("The", "A")); //11.Replace the word “The” with the word “A”.

System.***out***.println("Split string as 2 animal separates : 1st string is -- "+ str.substring(0, str.indexOf("dog")) + " -- and 2nd string is -- " + str.substring(str.indexOf("dog")) ); //12.Split the above string into two such that two animal names do not come together.

System.***out***.println("Animal names only : " + str.substring(str.indexOf("fox"), str.indexOf("fox")+3 ) + " " + str.substring(str.indexOf("dog"), str.indexOf("dog")+3 ) ); //13.Print the animal names alone separately from the above string.

System.***out***.println("Lowercase : "+ str.toLowerCase()); //14.Print the above string in completely lower case.

System.***out***.println("Uppercase : "+ str.toUpperCase()); //15.Print the above string in completely upper case.

}

}

**Lab Exercise No:**52

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

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

**Recommended duration:** *20 Mins*

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

**package** com.hsbc.lab4;

/\*

\* Handling NullPointerException

\* \*/

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

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

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

String str11=**null**;

**try**{

**if**(str11==**null**){

**throw** **new** NullPointerException("Null value");

}

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

System.***out***.println("12th index character is : "+ str.charAt(12));

System.***out***.println("Word 'is' is present : "+ str.contains("is")); // 2.Check whether the String contains the word “is”.

str += " and killed it";

System.***out***.println("Appended string : "+ str); //3.Add the string “and killed it” to the existing string.

System.***out***.println("String ends with 'dogs\*' : "+ str.endsWith("dogs")); // 4.Check whether the String ends with the word “dogs”.

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

String checkingString2 = "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG";

System.***out***.println("String is equal with (The quick brown Fox jumps over the lazy Dog) : "+ (str==checkingString)); //5.Check whether the String is equal to “The quick brown Fox jumps over the lazy Dog”.

System.***out***.println("String is equal with (THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG) : "+ (str==checkingString2)); //6.Check whether the String is equal to “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”.

System.***out***.println("Index of char 'a' is : "+ str.indexOf("a")); //7.Find the index position of the character “a”.

System.***out***.println("Last index of char 'e' is : "+ str.lastIndexOf("e")); //8.Find the last index position of the character “e”.

System.***out***.println("Length of string is "+ str.length()); //9.Find the length of the String.

System.***out***.println("String is matches with (The quick brown Fox jumps over the lazy Dog) : "+ (str.matches(checkingString))); //10.Check whether the String matches to “The quick brown Fox jumps over the lazy Dog”.

System.***out***.println("After replacing word 'The' with 'A' : "+ str.replace("The", "A")); //11.Replace the word “The” with the word “A”.

System.***out***.println("Split string as 2 animal separates : 1st string is -- "+ str.substring(0, str.indexOf("dog")) + " -- and 2nd string is -- " + str.substring(str.indexOf("dog")) ); //12.Split the above string into two such that two animal names do not come together.

System.***out***.println("Animal names only : " + str.substring(str.indexOf("fox"), str.indexOf("fox")+3 ) + " " + str.substring(str.indexOf("dog"), str.indexOf("dog")+3 ) ); //13.Print the animal names alone separately from the above string.

System.***out***.println("Lowercase : "+ str.toLowerCase()); //14.Print the above string in completely lower case.

System.***out***.println("Uppercase : "+ str.toUpperCase()); //15.Print the above string in completely upper case.

}

//Exception 1

**catch**(StringIndexOutOfBoundsException e)

{

e.printStackTrace();

}

**catch** (Exception e){

// General catch stataement for all exceptions

System.out.println(e);

}

}

**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*

**package** com.hsbc.pack;

**import** java.util.Scanner;

/\*

\* Program to find the square of each elements of the matrix.

\*/

**public** **class** MatrixSquare {

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

**try**{

**int** row, col, i, j;

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

System.***out***.println("Value of Rows and Columns of Matrix");

row = s.nextInt();

col = s.nextInt();

// Matrix creation

**int** first[][] = **new** **int**[row][col];

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

System.***out***.println("Elements of first Matrix");

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

**for** (j = 0; j < col; j++)

first[i][j] = s.nextInt();

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

{

**for** (j = 0; j < col; j++)

{ square[i][j] = first[i][j] \* first[i][j]; }

}

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

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

{

**for** (j = 0; j < col; j++)

System.***out***.print(square[i][j] + " ");

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.*

**Recommended duration:** *20 Mins*

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

**package** com.hsbc.lab4;

/\*

\* Handling all Exception mentioned below with respective Catch statement.

\* NegativeArraySizeException,ArrayIndexOutOfBoundsException,

\* StringIndexOutOfBoundsException, IndexOutOfBoundsException, NullPointerException,

\* ArithmeticException

\* \*/

**import** java.util.Scanner;

**public** **class** Solution54 {

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

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

**try**{

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

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

// NegativeArraySizeException can happen if l is smaller than 0:

**int**[] arr = **new** **int**[l];

**for**(**int** i=0;i<l;i++) {

arr[i] = sc.nextInt();

}

//IndexOutOfBoundsException

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

String n = "Hello Hello";

//StringIndexOutOfBoundsException

System.***out***.println(n.charAt(100));

String s = **null**;

s.toString();

**int** a = 6,b =0;

**int** c = a/b;

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

**int**[] arr=**new** **int**[-10];

**int** arr1[] = {1,2,3,4,5};

**int** element = arr1[6];

String s = **null**;

//NullPointerException

System.***out***.println(s.equals("blah"));

//ArithmeticException

**int** a=10;

**int** b=0;

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

}

**catch**(NegativeArraySizeException e){

e.printStackTrace();

}

**catch**(StringIndexOutOfBoundsException e){

e.printStackTrace();

}

**catch**(IndexOutOfBoundsException e){

e.printStackTrace();

}

**catch**(NullPointerException e){

e.printStackTrace();

}

**catch**(ArithmeticException e){

e.printStackTrace();

}

**catch**(Exception e){

e.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.*

**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.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Use finally block*

**package** com.hsbc.lab4;

/\*

\* Example of using finally block with try-catch block

\* \*/

**import** java.util.Scanner;

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

**int** var1,var2;

**public** Solution56(**int** var1, **int** var2) {

**super**();

**this**.var1 = var1;

**this**.var2 = var2;

}

**void** div(){

**double** res;

**try**{

res = **this**.var1/**this**.var2;

System.***out***.println("Division: "+res);

}

**catch**(ArithmeticException e){

e.printStackTrace();

}

}

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

**int** a,b;

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

System.***out***.println("Enter two numbers: ");

a = s.nextInt();

b = s.nextInt();

Solution56 obj = **new** Solution56(a,b);

**try**{

obj.div();

}

**catch**(ArithmeticException e){

e.printStackTrace();

}

**finally**{

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

obj=**null**; // setting null to object

}

s.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.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Use finally block*

**package** com.hsbc.lab4;

/\*

\* Defining try-catch and finally block in method of class.

\* \*/

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

**public** String example=**null**;

**public** String giveExample(){

**try**{

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

}

**catch**(NullPointerException e){

e.printStackTrace();

}

// This block will always return an object of String.

**finally**{

example += "value returned";

**return** example;

}

}

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

Solution57 obj = **new** Solution57();

// Calling method

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

}

}

**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.*

**Recommended duration:** *20 Mins*

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

**File-1)**

**package** com.hsbc.lab4;

/\*

\* defining AgeLimitException

\* \*/

**public** **class** AgeLimitException **extends** Exception{

**public** AgeLimitException(String s) {

**super**(s);

}

}

**File-2)**

**package** com.hsbc.lab4;

/\*

\* defining NameExistException

\* \*/

**public** **class** NameExistException **extends** Exception{

**public** NameExistException(String s) {

**super**(s);

}

}

**File-3)**

**package** com.hsbc.lab4;

/\*

\* defining NegativeAgeException

\* \*/

**public** **class** NegativeAgeException **extends** Exception{

**public** NegativeAgeException(String s) {

**super**(s);

}

}

Driver Class)

**package** com.hsbc.lab4;

/\*

\* Defining a User-defined Exception according to the Name and age value.

\* \*/

**import** com.hsbc.lab4.\*;

**import** java.util.\*;

**public** **class** Employee58{

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

**static** ArrayList<String> *names* = **new** ArrayList<String>();

**public** **void** getName()**throws** NameExistException{

System.***out***.print("Enter your name : ");

String name = s.next();

//checking if the name already exists or not

**if**(*names*.contains(name))

{ **throw** **new** NameExistException("Name already Exists"); }

**else** {

System.***out***.println("Hi " + name);

// adding name if it is not already in ArrayList

*names*.add(name);

}

}

**public** **void** getAge() **throws** NegativeAgeException,AgeLimitException{

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

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

**if**(age<0)

**throw** **new** NegativeAgeException("Can't have negative age!");

**else**{

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

{ **throw** **new** AgeLimitException("Age is out of limit!"); }

**else**

{ System.***out***.println("Your age is in range"); }

}

}

**public** **static** **void** main(String[] args) **throws** NameExistException,NegativeAgeException,AgeLimitException {

Employee58 obj = **new** Employee58();

**try**{

obj.getName();

}

**catch** (NameExistException e){

System.***out***.println("Exception occured!");

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

}

**try**{

obj.getAge();

}

**catch**(NegativeAgeException e){

System.***out***.println("Exception occured!");

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

}

**catch**(AgeLimitException e){

System.***out***.println("Exception occured!");

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

}

}

}