Problem-1

Develop a simple java application to calculate the tax for a particular employee based on his salary. Develop a class named “TaxCalculator“ with a method named calculateTax with the following method parameters, Variable Name Data Type empName String isIndian boolean empSal double This method should return a double taxAmount. The business logic for calculating the tax is as follows, this has to be implemented inside the method, If the employee is not a Indian The calculator should throw a CountryNotValidException If the employee name is null or empty The calculator should throw a EmployeeNameInvalidException If empSal is greater than one lakh and isIndian true taxAmount =empSal \*8/100 Otherwise If empSal is between 50K and 1lakh and isIndian true taxAmount =empSal \*6/100 Otherwise If empSal is between 30 and 50 Thousand and isIndian true taxAmount =empSal \*5/100 Otherwise If empSal is between 10 and 30 Thousand and isIndian true taxAmount =empSal \*4/100 Otherwise The calculator should throw a TaxNotEligibleException. Develop a main class CalculatorSimulator , implement the following logic in main method 1. Execute the calculateTax Method and print the tax amount as “Tax amount is “ 2. In case the calculateTaxMethod throws exceptions, this method needs to catch the appropriate exception print the stack trace and display the following messages, a. Country not valid: “The employee should be an Indian citizen for calculating tax” b. Employee name not valid: “The employee name cannot be empty” c. Not eligible for Tax calculation: “The employee does not need to pay tax” The following test cases to be executed, change the data in main method and run it and verify the output messages Test Cases Employee Name Employee Salary Is Indian Message Expected Test Case 1 Ron 34000 False The employee should be an Indian citizen for calculating tax. Test Case 2 Tim 1000 True The employee does not need to pay tax Test Case 3 Jack 55000 True Tax amount is 3300 Test Case 4 30000 True The employee name cannot be empty.

Code-

package pkg;

//Custom exception for invalid country

class CountryNotValidException extends Exception {

public CountryNotValidException(String message) {

super(message);

}

}

//Custom exception for invalid employee name

class EmployeeNameInvalidException extends Exception {

public EmployeeNameInvalidException(String message) {

super(message);

}

}

//Custom exception for non-eligibility of tax

class TaxNotEligibleException extends Exception {

public TaxNotEligibleException(String message) {

super(message);

}

}

class TaxCalculator {

String empName;

boolean isIndian;

double empSal;

double taxAmount;

// Method to calculate tax based on employee details

public void calculateTax(String empName, boolean isIndian, double empSal)

throws CountryNotValidException, EmployeeNameInvalidException, TaxNotEligibleException {

if (!isIndian) {

throw new CountryNotValidException("The employee should be an Indian citizen for calculating tax.");

}else if (empName == null) {

throw new EmployeeNameInvalidException("The employee name cannot be empty.");

}else if(empSal > 100000 && isIndian) {

taxAmount = empSal\*8/100; // 8% tax for salary > 1 lakh

System.out.println("Tax Amount is :"+ taxAmount);

} else if (empSal >= 50000 && empSal <= 100000 && isIndian) {

taxAmount = empSal\*6/100; // 6% tax for salary between 50k and 1 lakh

System.out.println("Tax Amount is :"+ taxAmount);

} else if (empSal >= 30000 && empSal < 50000 && isIndian) {

taxAmount = empSal\*5/100; // 5% tax for salary between 30k and 50k

System.out.println("Tax Amount is :"+ taxAmount);

} else if (empSal >= 10000 && empSal < 30000 && isIndian) {

taxAmount = empSal\*4/100; // 4% tax for salary between 10k and 30k

System.out.println("Tax Amount is :"+ taxAmount);

} else {

throw new TaxNotEligibleException("The employee does not need to pay tax.");

}

}

public static void main(String[] args) throws CountryNotValidException, EmployeeNameInvalidException, TaxNotEligibleException

{

// Create an instance of TaxCalculator

TaxCalculator tax = new TaxCalculator();

try {

tax.calculateTax("tim", true , 1000);

}catch (Exception e) {

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

}

}

}

Output-

The employee does not need to pay tax.

Problem-2

Create a program with a logic that throws the ArrayIndexOutOfBoundsException while accessing elements in an array.

Code-

package pkg;

public class ArrayAccess {

public static void main(String[] args) {

// Define an array of size 5

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

// Access an invalid index to cause ArrayIndexOutOfBoundsException

try {

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

System.out.println("Element at Index" + i + ":"+ arr[i]);

}

} catch (ArrayIndexOutOfBoundsException e) {

// Handling the exception

System.out.println("Error: Array index out of bounds.");

}

}

}

Output-

Element at Index0:1

Element at Index1:2

Element at Index2:3

Element at Index3:4

Element at Index4:5

Error: Array index out of bounds.