BYAGARI PAVAN PBL ID: J\_251890123

**Hands-on Assignments for Exception Handling**

1. **Write a program that takes as input the size of the array and the elements in the array. The program then asks the user to enter a particular index and prints the element at that index. Index starts from zero.**

**This program may generate Array Index Out Of Bounds Exception or**

**NumberFormatException Use exception handling mechanisms to handle this exception.**

**Sample Input and Output 1:**

**Enter the number of elements in the array 2**

**Enter the elements in the array 50**

**80**

**Enter the index of the array element you want to access 1**

**The array element at index 1 = 80**

**The array element successfully accessed Sample Input and Output 2:**

**Enter the number of elements in the array 2**

**Enter the elements in the array 50**

**80**

**Enter the index of the array element you want to access G**

**java.lang.ArrayIndexOutOfBoundsException Sample Input and Output 3:**

**Enter the number of elements in the array 2**

**Enter the elements in the array 30**

**j**

**java.lang.NumberFormatException**

import java.util.Scanner;

public class ArrayAccess {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

try {

// Step 1: Take size of array

System.out.println("Enter the number of elements in the array");

int n = Integer.parseInt(sc.nextLine()); // may throw NumberFormatException

int[] arr = new int[n];

// Step 2: Read elements

System.out.println("Enter the elements in the array"); for (int i = 0; i < n; i++) {

arr[i] = Integer.parseInt(sc.nextLine()); // may throw NumberFormatException

}

// Step 3: Read index

System.out.println("Enter the index of the array element you want to access"); int index = Integer.parseInt(sc.nextLine());

// Step 4: Access element

System.out.println("The array element at index " + index + " = " + arr[index]);

System.out.println("The array element successfully accessed");

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println(e); // prints java.lang.ArrayIndexOutOfBoundsException

} catch (NumberFormatException e) {

System.out.println(e); // prints java.lang.NumberFormatException

}

sc.close();

}

}

**Outputs**

**Case 1 (valid access)**

Enter the number of elements in the array 2

Enter the elements in the array 50

80

Enter the index of the array element you want to access 1

The array element at index 1 = 80

The array element successfully accessed

**Case 2 (invalid index)**

Enter the number of elements in the array 2

Enter the elements in the array 50

80

Enter the index of the array element you want to access 9

java.lang.ArrayIndexOutOfBoundsException

**Case 3 (invalid number input)**

Enter the number of elements in the array 2

Enter the elements in the array 30

j java.lang.NumberFormatException

1. **Write a class MathOperation which accepts 5 integers through command line. Create an array using these parameters. Loop through the array and obtain the sum and average of all the elements and display the result.**

**Various exceptions that may arise like ArithmeticException, NumberFormatException, and so on should be handled.**

public class MathOperation {

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

// Check if exactly 5 arguments are passed if (args.length != 5) {

throw new IllegalArgumentException("Please provide exactly 5 integer arguments.");

}

int[] numbers = new int[5]; int sum = 0;

// Convert arguments to integers (may throw NumberFormatException) for (int i = 0; i < 5; i++) {

numbers[i] = Integer.parseInt(args[i]);

sum += numbers[i];

}

// Calculate average (may throw ArithmeticException if divide by 0, though not here since 5 > 0)

int average = sum / numbers.length;

// Display results

System.out.println("Sum = " + sum);

System.out.println("Average = " + average);

} catch (NumberFormatException e) {

System.out.println("NumberFormatException: Please enter valid integers only.");

} catch (ArithmeticException e) {

System.out.println("ArithmeticException: " + e.getMessage());

} catch (IllegalArgumentException e) {

System.out.println("IllegalArgumentException: " + e.getMessage());

} catch (Exception e) {

// Catch any other unexpected exceptions System.out.println("Exception: " + e);

}

}

}

**Case 1 (Valid input)**

java MathOperation 10 20 30 40 50 Output:

Sum = 150

Average = 30

**Case 2 (Invalid number)**

java MathOperation 10 20 a 40 50

Output:

NumberFormatException: Please enter valid integers only.

**Case 3 (Less arguments)** java MathOperation 10 20 30 Output:

IllegalArgumentException: Please provide exactly 5 integer arguments.

1. **Write a Program to take care of Number Format Exception if user enters values other than integer for calculating average marks of 2 students. The name of the students and marks in 3 subjects are taken from the user while executing the program.**

**In the same Program write your own Exception classes to take care of Negative values and values out of range (i.e. other than in the range of 0-100)**

import java.util.Scanner;

// Custom Exception for negative values

class NegativeValueException extends Exception { public NegativeValueException(String message) {

super(message);

}

}

// Custom Exception for out-of-range values (>100) class OutOfRangeException extends Exception {

public OutOfRangeException(String message) { super(message);

}

}

public class StudentMarks {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

try {

for (int student = 1; student <= 2; student++) {

System.out.println("Enter name of Student " + student + ": "); String name = sc.nextLine();

int[] marks = new int[3]; int sum = 0;

System.out.println("Enter marks for 3 subjects (0-100): "); for (int i = 0; i < 3; i++) {

String input = sc.nextLine();

// May throw NumberFormatException int mark = Integer.parseInt(input);

// Custom exception checks if (mark < 0) {

throw new NegativeValueException("Negative marks not allowed: " + mark);

}

if (mark > 100) {

throw new OutOfRangeException("Marks cannot exceed 100: " + mark);

}

marks[i] = mark; sum += mark;

}

double average = sum / 3.0;

System.out.println("Average marks for " + name + " = " + average);

}

}

catch (NumberFormatException e) {

System.out.println("NumberFormatException: Please enter valid integer marks.");

}

catch (NegativeValueException e) {

System.out.println("NegativeValueException: " + e.getMessage());

}

catch (OutOfRangeException e) {

System.out.println("OutOfRangeException: " + e.getMessage());

}

catch (Exception e) {

System.out.println("Unexpected Exception: " + e);

}

finally { sc.close();

}

}

}

**Output:**

Enter name of Student 1: Arun

Enter marks for 3 subjects (0-100):

80

90

70

Average marks for Arun = 80.0

Enter name of Student 2:

Madhan

Enter marks for 3 subjects (0-100):

60

75

85

Average marks for Madhan = 73.33333333333333

1. **A student portal provides user to register their profile. During registration the system needs to validate the user should be located in India. If not the system should throw an exception.**

**Step 1: Create a user defined exception class named "InvalidCountryException". Step 2: Overload the respective constructors.**

**Step 3: Create a main class "UserRegistration", add the following method,**

**void registerUser (String username, String userCountry) with the below implementation**

**if userCountry is not equal to "India" throw a InvalidCountry Exception with the message "User Outside India cannot be registered"**

**if userCountry is equal to "India", print the message "User registration done successfully"**

**Invoke the method registerUser from the main method with the data specified and see how the program behaves.**

**Example1)**

**i/p: Mickey, US**

**o/p: InvalidCountry Exception should be thrown.**

**The message should be "User Outside India cannot be registered" Example2)**

**i/p:Mini, India**

**o/p: User registration done successfully**

// Step 1: Custom Exception Class

class InvalidCountryException extends Exception {

// Default constructor

public InvalidCountryException() {

super("User Outside India cannot be registered");

}

// Constructor with custom message

public InvalidCountryException(String message) { super(message);

}

}

// Step 2: Main Class

public class UserRegistration {

// Method to register user

void registerUser(String username, String userCountry) throws InvalidCountryException {

if (!userCountry.equalsIgnoreCase("India")) {

throw new InvalidCountryException("User Outside India cannot be registered");

} else {

System.out.println("User registration done successfully");

}

}

// Main Method

public static void main(String[] args) {

UserRegistration ur = new UserRegistration();

try {

// Example 1: Mickey, US ur.registerUser("Mickey", "US");

} catch (InvalidCountryException e) { System.out.println(e.getMessage());

}

try {

// Example 2: Mini, India ur.registerUser("Mini", "India");

} catch (InvalidCountryException e) { System.out.println(e.getMessage());

}

}

}

**Output Case 1:**

Input → Mickey, US

Output →

User Outside India cannot be registered

**Case 2:**

Input → Mini, India Output →

User registration done successfully

1. **Write a program to accept name and age of a person from the command prompt (passed as arguments when you execute the class) and ensure that the age entered is >=18 and 60.**

**Display proper error messages.**

**The program must exit gracefully after displaying the error message in case the arguments passed are not proper.**

**(Hint: Create a user defined exception class for handling errors.)**

// Custom Exception Class

class InvalidAgeException extends Exception { public InvalidAgeException(String message) {

super(message);

}

}

public class Person {

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

// Check if 2 arguments are passed if (args.length != 2) {

throw new IllegalArgumentException("Please provide exactly 2 arguments:

<Name> <Age>");

}

String name = args[0];

int age = Integer.parseInt(args[1]); // may throw NumberFormatException

// Validate age

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

throw new InvalidAgeException("Age must be between 18 and 60. Entered age: "

+ age);

}

// If valid

System.out.println("Registration successful!"); System.out.println("Name: " + name);

System.out.println("Age: " + age);

} catch (NumberFormatException e) {

System.out.println("Error: Age must be a valid integer.");

} catch (InvalidAgeException e) {

System.out.println("InvalidAgeException: " + e.getMessage());

} catch (IllegalArgumentException e) {

System.out.println("Error: " + e.getMessage());

} catch (Exception e) {

System.out.println("Unexpected Error: " + e);

}

}

}

Output:

Registration successful! Name: Arun

Age: 25