**1. Division with Exception Handling**

public class DivisionExample {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

try {

System.out.print("Enter first number: ");

int num1 = sc.nextInt();

System.out.print("Enter second number: ");

int num2 = sc.nextInt();

int result = num1 / num2;

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

} catch (ArithmeticException e) {

System.out.println("Error: Division by zero is not allowed.");

}

}

}

**2. Array Index Out Of Bounds Exception & String Index Out Of Bounds Exception**

public class ExceptionExamples {

public static void main(String[] args) {

// ArrayIndexOutOfBoundsException

try {

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

System.out.println(arr[5]);

} catch (ArrayIndexOutOfBoundsException e) {

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

}

// StringIndexOutOfBoundsException

try {

String str = "Hello";

System.out.println(str.charAt(10));

} catch (StringIndexOutOfBoundsException e) {

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

}

}

}

**3. Custom Exception (Invalid Age Exception)**

public class AgeCheck {

static void validateAge(int age) throws InvalidAgeException {

if (age < 18) {

throw new InvalidAgeException("Age must be 18 or above.");

} else {

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

}

}

public static void main(String[] args) {

try {

validateAge(15);

} catch (InvalidAgeException e) {

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

}

}

}

**4. File Handling with Exception**

public class FileReadExample {

public static void main(String[] args) {

try {

File file = new File("test.txt");

Scanner sc = new Scanner(file);

while (sc.hasNextLine()) {

System.out.println(sc.nextLine());

}

sc.close();

} catch (FileNotFoundException e) {

System.out.println("Error: File not found.");

}

}

}

**5. ArrayList Example**

public class ArrayListExample {

public static void main(String[] args) {

ArrayList<String> list = new ArrayList<>();

list.add("Apple");

list.add("Banana");

list.add("Cherry");

System.out.println("ArrayList before clearing: " + list);

list.clear();

System.out.println("ArrayList after clearing: " + list);

}

}

**6. TreeMap Example**

public class TreeMapExample {

public static void main(String[] args) {

TreeMap<Integer, String> employees = new TreeMap<>();

employees.put(103, "Ravi");

employees.put(101, "Anil");

employees.put(102, "Priya");

System.out.println("Employees in alphabetical order of IDs:");

for (Map.Entry<Integer, String> entry : employees.entrySet()) {

System.out.println(entry.getKey() + " -> " + entry.getValue());

}

}

}

### **7. Convert List to Array**

public class ListToArray {

public static void main(String[] args) {

List<String> list = new ArrayList<>();

list.add("One");

list.add("Two");

list.add("Three");

String[] arr = list.toArray(new String[0]);

System.out.println("Converted Array:");

for (String s : arr) {

System.out.println(s);

}

}

}