Task 6
A) Built-in Exceptions Using Separate Try–Catch Blocks

Aim

To demonstrate ArrayIndexOutOfBoundsException, StringIndexOutOfBoundsException, NullPointerException and NumberFormatException using separate try–catch blocks.

Algorithm

1. Start the program.
2. Create an array of size 3 and access index 5.
3. Catch ArrayIndexOutOfBoundsException.
4. Create a string "JAVA" and access index 10.
5. Catch StringIndexOutOfBoundsException.
6. Declare a string as null and call length().
7. Catch NullPointerException.
8. Convert "ABC123" using Integer.parseInt().
9. Catch NumberFormatException.
10. Stop the program.

Program

```
public class BuiltInExceptions {
   public static void main(String[] args) {
      // 1. ArrayIndexOutOfBoundsException
      try {
         int arr[] = {10, 20, 30};
         System.out.println(arr[5]);
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("ArrayIndexOutOfBoundsException: " + e.getMessage());
      }

      // 2. StringIndexOutOfBoundsException
      try {
         String str = "JAVA";
         System.out.println(str.charAt(10));
      } catch (StringIndexOutOfBoundsException e) {
         System.out.println("StringIndexOutOfBoundsException: " + e.getMessage());
      }

      // 3. NullPointerException
      try {
         String s = null;
         System.out.println(s.length());
      } catch (NullPointerException e) {
         System.out.println("NullPointerException: " + e.getMessage());
      }
      // 4. NumberFormatException
```

```
        try {
            String value = "ABC123";
            int num = Integer.parseInt(value);
            System.out.println(num);
        } catch (NumberFormatException e) {
            System.out.println("NumberFormatException: " + e.getMessage());
        }
    }
}
```

Sample Input

Array size = 3
Array elements = 10 20 30
Access index = 5

String = "JAVA"
Access character index = 10

String reference = null
Operation = length()

String value = "ABC123"
Conversion = Integer.parseInt()

Output

ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3
StringIndexOutOfBoundsException: String index out of range: 10
NullPointerException: Cannot invoke "String.length()" because the object is null
NumberFormatException: For input string: "ABC123"

Result

Thus, the built-in exceptions were successfully demonstrated using separate try–catch blocks.

B) User-Defined Exception

AIM

To define and handle a user-defined exception when age is less than 18.

ALGORITHM

1. Start the program.
2. Create a custom exception class named InvalidAgeException.
3. Read age from user.
4. If age < 18, throw InvalidAgeException.
5. Catch and display the exception message.
6. Stop the program

PROGRAM

```
import java.util.Scanner;

// User-defined Exception
class InvalidAgeException extends Exception {
   public InvalidAgeException(String message) {
      super(message);
   }
}

public class UserDefinedException {

   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

      try {
         System.out.print("Enter age: ");
         int age = sc.nextInt();

         if (age < 18) {
            throw new InvalidAgeException("Age must be 18 or above");
         }

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

      sc.close();
   }
}
```

INPUT
Enter age: 16

OUTPUT
Enter age: 16
InvalidAgeException: Age must be 18 or above

RESULT
Thus, the user-defined exception was successfully created and handled using try–catch block.