**1) Ramesh is developing a student management system for a university. In this system, you have a Student class to represent student information. You are Asked to Help Ramesh to handle exception which can be occurred into program according to following Scenarios:**

**1)class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor.**

**2)If the age of the student is not between 15 and 21 then generate a user-defined exception "AgeNotWithinRangeException".**

**3)If a name contains numbers or special symbols, raise exception "NameNot ValidException". Define the two exception classes**

**OUTPUT:**

**Case 1: Valid Student**

Student{rollNo=101, name='John Doe', age=18, course='Computer Science'}

**Case 2: Invalid Age**

Error: Age must be between 15 and 21.

**Case 3: Invalid Name**

Error: Name contains invalid characters. Only alphabets are allowed.

**2.** **Create a class Voter(voterId, name, age) with parameterized constructor. The parameterized constructor should throw a checked exception if age is less than 18. The message of exception is "invalid age for voter "  
  
OUTPUT:  
  
Case 1: Valid Voter**

Enter voter ID: 101

Enter voter name: John Doe

Enter voter age: 25

Voter registered successfully: Voter{voterId=101, name='John Doe', age=25}

**Case 2: Underage Voter**

Enter voter ID: 102

Enter voter name: Jane Doe

Enter voter age: 16

Error: Invalid age for voter

**Case 3: Invalid Input**

Enter voter ID: 103

Enter voter name: Alice

Enter voter age: abc

Error: Invalid input. Please try again.

**3.** **Store name of weekdays in an array (starting from "Sunday" at 0 index). Ask day position from user and print day name. Handle array index out of bound exception and give proper message if user enters day index outside range (0-6)**

**OUTPUT:**

**Case 1: Valid Day Index**

Enter the day position (0-6): 3

The day is: Wednesday

**Case 2: Invalid Day Index**

Enter the day position (0-6): 7

Error: Invalid day index. Please enter a number between 0 and 6.

**Case 3: Non-Numeric Input**

Enter the day position (0-6): abc

Error: Invalid input. Please enter a numeric value.

**4Create a HashMap where keys are student names (strings) and values are their corresponding grades (integers). Create methods to add a new student, remove a student, and Display up a student's grade by name.**

**OUTPUT:**

**Case 1: Add a Student**

Menu:

1. Add Student

2. Remove Student

3. Display Student's Grade

4. Exit

Enter your choice: 1

Enter student name: John

Enter student grade: 85

Student added: John with grade 85

**Case 2: Display a Student's Grade**

Enter your choice: 3

Enter student name to display grade: John

Grade of John: 85

**Case 3: Remove a Student**

Enter your choice: 2

Enter student name to remove: John

Student removed: John

**Case 4: Invalid Name**

Enter your choice: 3

Enter student name to display grade: Jane

Student not found: Jane

**Case 5: Exit Program**

Enter your choice: 4

Exiting...

**5.Use Collection Classes to store Integers.Create some methods for following functionalities a. Include functions for pushing elements onto the stack. b. popping elements from the stack. c. checking if the stack is empty.**

**OUTPUT:**

**Case 1: Initial Check**

The stack is empty.

**Case 2: Push Elements**

Pushed element: 10

Pushed element: 20

Pushed element: 30

**Case 3: Check If Empty**

The stack is not empty.

**Case 4: Pop Elements**

Popped element: 30

Popped element: 20

**Case 5: Check After Popping**

The stack is not empty.

**Case 6: Pop Remaining Element**

Popped element: 10

The stack is empty. No element to pop.