

## Practice on nested try catch block

1. Design a program for calculating the monthly mortgage payment based on user input for loan amount, interest rate, and loan term. Handle arithmetic-related exceptions such as `ArithmeticException` using nested try-catch blocks.

**Sample Input:** Loan amount, interest rate, loan term (in years)

**Sample Output:** Monthly mortgage payment or an error message if the calculation encounters a divide-by-zero error due to invalid input or exceptional circumstances.

# Practice on throw keyword

2. Suppose you are developing a social media platform where users need to be at least 18 years old to create an account. Upon registration, users are required to input their age. Write a Java program to verify the age of a user during the registration process. If the age provided is less than 18, throw a `SecurityException` with a message indicating that the user is underage.

**Sample Input:**

Enter age: 15

**Sample Output:**

`SecurityException: User is underage (age: 15)`

## THROW KEYWORD

Write a Java program to calculate the grade of a student based on their marks. If the marks provided are negative or greater than 100, throw an `IllegalArgumentException`. The program should use the `throws` keyword in the method signature to indicate that it may throw an `IllegalArgumentException`.

**Sample Input:**

**Marks: 85**

**Sample Output:**

**Grade: B**

```
public class GradeCalculator {  
    public static char calculateGrade(int marks) throws IllegalArgumentException {  
        if (marks < 0 || marks > 100) {  
            throw new IllegalArgumentException("Marks should be between 0 and 100");  
        }  
        if (marks >= 90) {  
            return 'A';  
        } else if (marks >= 80) {  
            return 'B';  
        } else if (marks >= 70) {  
            return 'C';  
        } else if (marks >= 60) {  
            return 'D';  
        } else {  
            return 'F';  
        }  
    }  
}  
public static void main(String[] args) {  
    int marks = 85;  
    try {  
        char grade = calculateGrade(marks);  
        System.out.println("Grade: " + grade);  
    } catch (IllegalArgumentException e) {  
        System.out.println(e.getMessage());  
    }  
}}
```

## USER DEFINED EXCEPTION

**Write a Java program to simulate submitting a product review online. The program should prompt the user to enter their name, email, product rating (out of 5), and review text. If the review text provided by the user is empty or too short, throw an `InvalidReviewException`.**

**The `InvalidReviewException` should be thrown if the review text contains less than 10 characters.**

**You can assume that the user inputs are entered via the console. This question requires you to design a Java class for the product review submission process, including methods for validating the review text and handling the exception appropriately.**