1. Scenario:

You are working on a simple program to help users practice multiplication. The program will generate random multiplication questions for the user to answer. The user can continue answering questions until they choose to stop.

Requirements:

1. The program should generate two random integers between 1 and 10 for the multiplication question.
2. The program should prompt the user with a multiplication question and ask for the answer.
3. The program should check if the user's answer is correct or incorrect and provide feedback.
4. The loop should continue to ask new questions until the user decides to exit by entering a specific command (e.g., entering 0 to exit).

Task:

Write a Java program that implements the above scenario using a while loop.

Sample Output:

Welcome to the Multiplication Practice Program!

What is 3 * 7?

Enter your answer (or 0 to exit): 21

Correct!

What is 5 * 2?

Enter your answer (or 0 to exit): 9

Incorrect. The correct answer is 10.

What is 6 * 8?

Enter your answer (or 0 to exit): 0

Thank you for practicing multiplication!

2. Scenario:

You are developing a simple guessing game. In this game, the program will generate a random number between 1 and 100, and the user has to guess the number. The game will continue to prompt the user to guess until they guess the correct number.

Requirements:

1. The program should generate a random number between 1 and 100.
2. The program should prompt the user to guess the number.
3. The program should check if the user's guess is too high, too low, or correct and provide feedback.
4. If the user's guess is correct, the program should display a congratulatory message and end.
5. If the user's guess is incorrect, the program should continue to prompt the user for another guess.

Task:

Write a Java program that implements the above scenario using a do-while loop.

Sample Output:

Welcome to the Guessing Game!

I have selected a number between 1 and 100.

Can you guess what it is?

Enter your guess: 50

Too low!

Enter your guess: 75

Too high!

Enter your guess: 62

Congratulations! You guessed the correct number!

3. Scenario:

Sum of First n Natural Numbers

Task: Create a console application to calculate the sum of the first n natural numbers where n is provided by the user.

Requirements:

1. Prompt the user to enter a positive integer n.
2. Use a for loop to calculate the sum of the first n natural numbers.
3. Display the sum.

Sample Output:

Enter a positive integer: 5

Sum of first 5 natural numbers: 15

4. Scenario:

Factorial Calculation

Task: Create a console application to calculate the factorial of a given number n where n is provided by the user.

Requirements:

1. Prompt the user to enter a non-negative integer n.
2. Use a for loop to calculate the factorial of n.
3. Display the factorial.

Sample Output:

Enter a non-negative integer: 5

Factorial of 5 is 120

5. Scenario:

Pyramid Pattern of Stars

Task: Create a console application to print a pyramid pattern of stars with n levels where n is provided by the user.

Requirements:

1. Prompt the user to enter a positive integer n.
2. Use nested for loops to print the pyramid pattern.
3. Display the pattern.

Sample Output:

Enter the number of levels for the pyramid: 5

```
    *
   ***
  *****
 *******
*********
```

6. Scenario:

Multiplication Table

Task: Create a console application to print the multiplication table up to n x n where n is provided by the user.

Requirements:

1. Prompt the user to enter a positive integer n.
2. Use nested for loops to generate the multiplication table.
3. Display the table.

Sample Output:

Enter the size of the multiplication table: 5

Multiplication Table:

```
1  2  3  4  5
2  4  6  8 10
3  6  9 12 15
4  8 12 16 20
5 10 15 20 25
```

7. Scenario:

Pascal's Triangle

Task: Create a console application to print Pascal's Triangle up to n rows where n is provided by the user.

Requirements:

1. Prompt the user to enter a positive integer n.
2. Use nested for loops to generate and print Pascal's Triangle.

3. Display the triangle.

Sample Output:

Enter the number of rows for Pascal's Triangle: 5

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

8. Scenario:

Skipping Even Numbers

Task: Create a console application to print all odd numbers from 1 to n, skipping even numbers using the continue statement.

Requirements:

1. Prompt the user to enter a positive integer n.
2. Use a for loop to iterate from 1 to n.
3. Use a continue statement to skip even numbers.
4. Display the odd numbers.

Sample Output:

Enter a positive integer: 10

Odd numbers from 1 to 10:

1 3 5 7 9

9. Scenario:

Finding the First Divisible Number in a Range

Task: Create a console application to find the first number in a range from 1 to n that is divisible by a given number d.

Requirements:

1. Prompt the user to enter the upper limit n.
2. Prompt the user to enter the divisor d.
3. Use a for loop to iterate from 1 to n.
4. Use a break statement to stop the loop once a divisible number is found.
5. Display the number if found, or a message indicating none are divisible by d.

Sample Output:

Enter the upper limit: 10

Enter the divisor: 3

The first number divisible by 3 is 3.

10. Scenario:

User Authentication

Task: Create a console application that repeatedly prompts the user to enter a password until the correct password is entered. Use a break statement to exit the loop once the correct password is entered.

Requirements:

1. Define a hard-coded password.
2. Prompt the user to enter the password.
3. Use a while loop to repeatedly prompt the user until the correct password is entered.
4. Use a break statement to exit the loop once the correct password is entered.
5. Display a success message upon entering the correct password.

Sample Output:

Enter password: wrongpassword

Incorrect password. Try again.

Enter password: secret123

Password accepted.