

## Assignment 4: CONTROL STRUCTURES REPETITION

Program 1: Menu-Driven Calculator Using Loops( Use while or do while loop)

Program 2: TO print a pattern using nested for loops.

```

          1
        2   2
      3   3   3
    4   4   4   4
  5   5   5   5   5

```

Objective:

Program1:

In this assignment, you will create a menu-driven calculator that allows users to repeatedly perform arithmetic operations (addition, subtraction, multiplication, and division) on two numbers. The program will continue until the user chooses to exit. This project will help strengthen your ability to handle user inputs, implement control structures like loops, and use switch-case statements effectively.

Problem Description:

You will write a C++ program that:

1. Continuously prompts the user to input two numbers and an operator ( `+` , `-` , `\*` , `/` ).
2. Performs the corresponding arithmetic operation using a **switch-case** structure.
3. Displays the result of the calculation.
4. Asks the user whether they want to perform another calculation or exit the program.
5. If the user chooses to exit, the program should terminate gracefully.

## Requirements:

### 1. Input:

- Two numbers entered by the user.
- An operator ( $+$ ,  $-$ ,  $*$ ,  $/$ ) chosen by the user.
- A choice to continue or exit.

### 2. Output:

- The result of the arithmetic operation.
- A message asking if the user wants to continue or exit.

### 3. Control Structures:

- Use a do-while loop or a while loop to allow the user to repeatedly perform calculations until they choose to exit.
- Use a switch-case structure to handle the different arithmetic operations ( $+$ ,  $-$ ,  $*$ ,  $/$ ).

## Instructions:

### 1. Menu Display:

- Begin by displaying a simple menu with options for:
  - Addition ( $+$ )
  - Subtraction ( $-$ )
  - Multiplication ( $*$ )
  - Division ( $/$ )
- You can display the menu again after each operation or ask if the user wants to perform another calculation.

### 2. User Input:

- Prompt the user to enter two numbers.
- Prompt the user to choose one of the arithmetic operators.
- Validate the input to ensure that division by zero does not occur.

### 3. Perform Calculations:

- Based on the operator input by the user, perform the appropriate operation using a switch-case structure:

- Addition for `+`
- Subtraction for `-`
- Multiplication for `\*`
- Division for `/` (Ensure to handle division by zero carefully).

### 4. Display Results:

- After performing the calculation, display the result in a user-friendly format.

### 5. Loop Structure:

- After each calculation, ask the user if they want to perform another calculation or exit.
- Use a loop structure (like `while` or `do-while`) to continue or terminate the program based on the user's choice.

### 6. Exit Condition:

- If the user chooses to exit, display a goodbye message and terminate the program.

### Instructions for Program 2:

#### Program Structure:

The program must include everything in the main() function.

Initialize an integer variable "rows" to define the number of rows for the pyramid.

Use nested loops to create the pattern:

The outer loop will control the number of rows.

Two inner loops will be used:

One for printing leading spaces (indentation).

Another for printing the numbers in each row.

Steps to Follow:

Step 1: Initialize the Number of Rows

Declare an integer variable rows inside the main() function.

b. Step 2: Use a Loop to Print Each Row

Use an outer for loop to iterate through each row. The loop variable 'i' will represent the current row number.

c. Step 3: Print Leading Spaces

For each row, print the appropriate number of leading spaces before printing the numbers.

Use another for loop to print spaces. The number of spaces will decrease as you move down the rows, which creates the pyramid shape.

Use " " (4 spaces) for consistent spacing.

d. Step 4: Print Numbers in Each Row

After printing the leading spaces, print the row number i multiple times for each row.

The number of times the row number is printed corresponds to the row number (e.g., row 3 will print 3 three times).

Use " " (7 spaces) between the numbers to ensure proper alignment.

e. Step 5: Print a Newline

After printing the numbers for each row, print a newline (endl) to move to the next row.

Compilation and Execution:

Compile the program using a C++ compiler (e.g., g++) and run it.

Verify that the output matches the expected pyramid pattern.

Deliverables:

The source code of your C++ program.

A screenshot showing the program's output for 5 rows.

A short explanation (2-3 sentences) describing how the nested loops control the printing of spaces and numbers in the pyramid.

### Submission Instructions:

1. Ensure your code is well-documented with comments explaining each section.
2. Test your program thoroughly with different input values to ensure all scenarios are handled correctly.
3. Submit your 2programs code in a ``.cpp` file and screenshot to blackboard and cpp file to git.

Naming conventions: Assignment4\_Lastname1.cpp, Assignment4\_lastname2.cpp

1. Correctness: Does the program perform all calculations correctly?
2. Code Structure: Is the program well-organized and easy to read? Are proper control structures used?
3. User Interaction: Is the program user-friendly with clear prompts and output?

Good luck, and happy coding!

Sample output:

Program 2:

```

    1
  2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

## Program1:

```
Welcome to the Menu-Driven Calculator!
Enter the first number: 12
Enter the second number: 20
-----
Please choose the operation you'd like to perform:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
-----
Choose an operation (+, -, *, /): +
Result: 12 + 20 = 32
Do you want to perform another calculation? (y/n): y
Enter the first number: 20
Enter the second number: 40
-----
Please choose the operation you'd like to perform:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
-----
Choose an operation (+, -, *, /): |
```

```
Do you want to perform another calculation? (y/n): y
Enter the first number: 20
Enter the second number: 40
-----
Please choose the operation you'd like to perform:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
-----
Choose an operation (+, -, *, /): /
Result: 20 / 40 = 0.5
Do you want to perform another calculation? (y/n): y
Enter the first number: 30
Enter the second number: 5
-----
Please choose the operation you'd like to perform:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
-----
Choose an operation (+, -, *, /): *
Result: 30 * 5 = 150
Do you want to perform another calculation? (y/n): n
Thank you for using the calculator. Goodbye!
```

```
Welcome to the Menu-Driven Calculator!
Enter the first number: 10
Enter the second number: 0
-----
Please choose the operation you'd like to perform:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
-----
Choose an operation (+, -, *, /): /
ERROR!
Error: Division by zero is not allowed!
Do you want to perform another calculation? (y/n):
```