

1. C++ Basics

Problem Statement 1: **const** and **#define**

Write a C++ program to demonstrate the use of **const** and **#define** for defining constant values.

The program should:

1. Define the value of **PI** using **#define**.
2. Declare a constant integer **DAYS_IN_WEEK** using **const** and assign it the value **7**.
3. Display the values of **PI** and **DAYS_IN_WEEK**.
4. Declare a floating-point variable **radius** and assign it the value **5.0**.
5. Calculate the area of a circle using **PI** and **radius**.
6. Display the radius and calculated area.
7. Include a commented example showing that a **const** variable cannot be changed after initialization.

Expected Output:

```
Value of PI using #define: 3.14159
Days in a week using const: 7
Area of circle with radius 5 is: 78.5397
```

Problem Statement 2: Explicit Type Casting

Write a C++ program that:

1. Declares two integer variables **a = 5** and **b = 2**.
2. Declares a third variable **c** of type **float**.
3. Uses **explicit type casting** to convert one of the integer variables to **float** before performing the division.
4. Stores the division result in variable **c**.
5. Displays the value of **c**.

Expected Output

```
2.5
```

2. Input and Output

Problem Statement 1: – Student Information

Write a C++ program that takes the student's **name, age, and marks** as input and displays them using **cout**.

Problem Statement 2: – Calculate Total

Write a C++ program that takes the prices of three items from the user, calculates the total, and displays the result.

Problem Statement 3: – Calculate Rectangle Perimeter

Write a C++ program that takes the length and width of a rectangle and calculates:

$$\text{Perimeter} = 2 \times (\text{Length} + \text{Width})$$

Display the result using `cout`.

Problem Statement 4: – Calculate Average

Write a C++ program that takes marks of three subjects, calculates the average, and displays the result.

Problem Statement 5: – Student Age

Write a C++ program that asks the user to enter their age and displays:

Your age is: 20

Problem Statement 6: – Two Numbers

Write a C++ program that takes two integers from the user and displays their:

- Sum
 - Difference
 - Product
-

Problem Statement 7: – Student Marks

Write a C++ program that takes marks of three subjects and displays the total marks.

Problem Statement 8: – Rectangle

Write a C++ program that takes the length and width of a rectangle and calculates its area.

Formula:

$$\text{Area} = \text{Length} \times \text{Width}$$

Problem Statement 9: – Student Information

Write a C++ program that takes the following information from the user:

- Name
- Age
- Department
- Marks

Then display all the information using `cout`.

Problem Statement 10: – `endl`

Write a C++ program that displays your:

- Name
- Department
- Semester

Use `endl` to print each item on a separate line.

Problem Statement 11: – `setw`

Write a C++ program to display the following table using `setw(10)`:

Number	Cube
2	8
3	27
4	64
5	125

Problem Statement 12: – Both Manipulators

Write a C++ program that uses both `endl` and `setw()` to display a simple student information table containing:

- Roll No.
- Name
- Marks

3. Conditional Statements

Problem Statement 1: Positive, Negative or Zero - if statement

Write a C++ program that takes a number as input and checks whether it is positive, negative, or zero. Display the appropriate message based on the entered number.

Hint: Use an if-else-if ladder.

Problem Statement 2: Check Even or Odd - if statement

Write a C++ program that takes an integer as input and determines whether it is even or odd. Display an appropriate message based on the result.

Hint: Use the modulus operator (%) to check the remainder when the number is divided by 2.

Problem Statement 3: Check Eligibility Based on Marks - if statement

Write a C++ program that takes a student's marks as input. If the marks are 50 or above, display "Eligible"; otherwise, display "Not Eligible".

Hint: Use an if-else statement to check the marks.

Problem Statement 4: Student Grade Calculator - if statement

Write a C++ program that takes a student's marks as input and displays the corresponding grade according to the following criteria:

- **A:** 80–100 marks
- **B:** 70–79 marks
- **C:** 60–69 marks
- **D:** 50–59 marks
- **Fail:** Below 50 marks

If the entered marks are less than 0 or greater than 100, display "**Invalid Marks**".

Hint: Use an **if-else-if** ladder to check the marks and assign the appropriate grade. First, validate that the marks are between 0 and 100.

Problem Statement 5: Find the Larger Number - if statement

Write a C++ program that takes two numbers as input and compares them. Display the larger number. If both numbers are equal, display "Both numbers are equal."

Hint: Use an if-else-if ladder to compare the two numbers and handle all three possible cases.

Problem Statement 6: Login Authentication System - if statement

Write a C++ program that takes two inputs: a user verification flag and a password verification flag. Display "Login Successful" only if both flags indicate that the username and password are correct. Otherwise, display "Login Failed".

Hint: Use nested if statements to check the user verification flag first and then the password verification flag.

Problem Statement 7: Scholarship Eligibility Checker - if statement

Write a C++ program that takes a student's marks and family monthly income as input. The student is eligible for a scholarship only if both of the following conditions are met:

- The student's marks are 80 or above.
- The family's monthly income is below Rs. 60,000.

Display "Scholarship Awarded" if the student meets both conditions. Otherwise, display "Not Eligible for Scholarship".

Hint: Use nested if statements to check the marks first and then the family income.

Course Registration

Problem Statement 8: Student Registration Eligibility - if statement

Write a C++ program that takes three inputs to check whether a student can register:

- **Admission Status:** Admission is confirmed.
- **Fee Status:** Required fee has been paid.
- **Document Verification Status:** All required documents have been verified.

Display "**Registration Allowed**" only if all three requirements are satisfied. Otherwise, display "**Registration Not Allowed**".

Hint: Use nested **if** statements with appropriate **else** branches to check each requirement one by one.

Problem Statement 9: Day Type Checker Using **switch**

Write a C++ program that asks the user to enter a day number from 1 to 7. Use a **switch** statement to determine whether the entered day is a Weekday or a Weekend.

- If the user enters a number from 1 to 5, display **Weekday**.
- If the user enters 6 or 7, display **Weekend**.
- If the user enters any number outside the range of 1 to 7, display **Invalid day number**.

Example Output 1:

```
Enter day number (1-7): 3
Weekday
```

Example Output 2:

```
Enter day number (1-7): 7
Weekend
```

Example Output 3:

```
Enter day number (1-7): 9
Invalid day number
```

Problem Statement 10: Simple Calculator Using **switch**

Write a C++ program that asks the user to enter two numbers of type **double** and an arithmetic operator (+, -, *, /, or %). Use a **switch** statement to perform the selected arithmetic operation and display the result.

Requirements

- If the operator is +, add the two numbers.
- If the operator is -, subtract the second number from the first.
- If the operator is *, multiply the two numbers.
- If the operator is /, divide the first number by the second. If the second number is zero, display **Cannot divide by zero**.
- If the operator is %, convert both numbers to integers and calculate the remainder. If the converted second number is zero, display **Cannot calculate remainder by zero**.
- If the user enters an invalid operator, display **Invalid operator**.

Example Output 1: Addition

```
Enter first number: 10.5
Enter operator (+, -, *, /, %): +
Enter second number: 5.2
Result = 15.7
```

Example Output 2: Decimal Division

```
Enter first number: 10
Enter operator (+, -, *, /, %): /
Enter second number: 4
Result = 2.5
```

Example Output 3: Division by Zero

```
Enter first number: 10
Enter operator (+, -, *, /, %): /
Enter second number: 0
Cannot divide by zero
```

Example Output 4: Modulus

```
Enter first number: 10
Enter operator (+, -, *, /, %): %
Enter second number: 3
Result = 1
```

Example Output 5: Invalid Operator

```
Enter first number: 10
Enter operator (+, -, *, /, %): &
Enter second number: 5
Invalid operator
```

Note: Use `double` variables for both inputs. The modulus (%) operator works only with integers in C++, so convert the operands to integers for this operation. Any decimal parts will be discarded during conversion.

Problem Statement 11: Vowel Checker Using `switch`

Write a C++ program that asks the user to enter a single character. Use a `switch` statement to check whether the entered character is a vowel (`a`, `e`, `i`, `o`, or `u`). The program should recognize both lowercase and uppercase vowels.

Requirements

- If the character is `a`, `e`, `i`, `o`, or `u`, display `Vowel`.
- If the character is `A`, `E`, `I`, `O`, or `U`, display `Vowel`.
- For any other character, display `Not a vowel`.

Example Output 1: Lowercase Vowel

```
Enter a character: e
Vowel
```

Example Output 2: Uppercase Vowel

```
Enter a character: U
Vowel
```

Example Output 3: Consonant

```
Enter a character: b
Not a vowel
```

Example Output 4: Digit

```
Enter a character: 5
Not a vowel
```

Problem Statement 12: Student Management System Using `switch`

Write a C++ program to display a menu for a simple Student Management System. Ask the user to enter a choice from the menu and use a `switch` statement to perform the corresponding action.

Requirements

Display the following menu:

1. Student Profile
2. Semester Result
3. Fee Status
4. Exit

- If the user enters `1`, display `Displaying student profile.`
- If the user enters `2`, display `Displaying semester result.`
- If the user enters `3`, display `Displaying fee status.`
- If the user enters `4`, display `Exiting the program.`
- If the user enters any other number, display `Invalid choice.`

Example Output 1: Student Profile

```
--- Student Management System ---
1. Student Profile
2. Semester Result
3. Fee Status
4. Exit
Enter your choice: 1
Displaying student profile.
```

Example Output 2: Fee Status

```
--- Student Management System ---
1. Student Profile
2. Semester Result
3. Fee Status
4. Exit
Enter your choice: 3
Displaying fee status.
```

Example Output 3: Invalid Choice

```
--- Student Management System ---
1. Student Profile
2. Semester Result
3. Fee Status
4. Exit
Enter your choice: 5
Invalid choice.
```

Problem Statement 13: Grade Evaluation Using `switch`

Write a C++ program that asks the user to enter a letter grade (A, B, C, D, or F). Use a `switch` statement to display the corresponding performance message.

Requirements

1. If the grade is A, display `Excellent Performance`.
2. If the grade is B, display `Very Good Performance`.
3. If the grade is C, display `Good Performance`.
4. If the grade is D, display `You Passed`.
5. If the grade is F, display `You Failed`.
6. If the user enters any other character, display `Invalid grade`.
7. Modify the program to accept both uppercase and lowercase grades (e.g., A and a).
8. Add grade E and display `Needs Improvement`.
9. Test the program with valid and invalid grades to verify the output.

Sample Output 1

```
Enter your grade (A, B, C, D, F): A
Excellent Performance
```

Sample Output 2

```
Enter your grade (A, B, C, D, F): b
Very Good Performance
```

Sample Output 3

```
Enter your grade (A, B, C, D, F): X
Invalid grade
```

Problem Statement 14: Number of Days in a Month Using `switch`

Write a C++ program that asks the user to enter a month number (1–12). Use a `switch` statement to display the number of days in the corresponding month.

Requirements

1. If the user enters 1, display `31 days` for January.
2. If the user enters 2, display `28 days` for February.
3. If the user enters 3, display `31 days` for March.
4. If the user enters 4, display `30 days` for April.
5. If the user enters 5, display `31 days` for May.

6. If the user enters 6, display 30 days for June.
7. If the user enters 7, display 31 days for July.
8. If the user enters 8, display 31 days for August.
9. If the user enters 9, display 30 days for September.
10. If the user enters 10, display 31 days for October.
11. If the user enters 11, display 30 days for November.
12. If the user enters 12, display 31 days for December.
13. If the user enters a number outside the range of 1–12, display Invalid month number.

Note: Assume February has 28 days. Leap years are not considered.

Sample Output 1

```
Enter month number (1-12): 2
February has 28 days.
```

Sample Output 2

```
Enter month number (1-12): 4
April has 30 days.
```

Sample Output 3

```
Enter month number (1-12): 15
Invalid month number
```

Problem Statement 15: ATM Menu System Using switch

Difficulty Level: Advanced

Write a C++ program that simulates a simple ATM system. Display a menu that allows the user to check their account balance, deposit money, withdraw money, or exit the program. Use a **switch** statement to process the user's selection.

Requirements

1. **Balance Inquiry:** Display the current account balance.
2. **Deposit:** Ask the user to enter an amount and add it to the account balance. The amount must be greater than zero.
3. **Withdrawal:** Ask the user to enter an amount and subtract it from the account balance. The amount must be greater than zero and must not exceed the available balance.
4. **Exit:** Display Thank you for using the ATM. and terminate the program.
5. **Invalid Choice:** If the user enters an invalid menu option, display Invalid choice. Please try again.
6. Initialize the account balance with a starting amount of Rs. 10,000.

7. Display the menu again after each transaction until the user selects Exit.

Sample Output 1: Balance Inquiry

```
--- ATM Menu ---  
1. Balance Inquiry  
2. Deposit  
3. Withdrawal  
4. Exit  
  
Enter your choice: 1  
Current Balance: Rs. 10000
```

Sample Output 2: Deposit

```
Enter your choice: 2  
Enter deposit amount: 2000  
Deposit successful.  
Updated Balance: Rs. 12000
```

Sample Output 3: Withdrawal

```
Enter your choice: 3  
Enter withdrawal amount: 3000  
Withdrawal successful.  
Updated Balance: Rs. 9000
```

Sample Output 4: Insufficient Balance

```
Enter your choice: 3  
Enter withdrawal amount: 15000  
Insufficient balance.
```

Sample Output 5: Exit

```
Enter your choice: 4  
Thank you for using the ATM.
```