Week 2 - Conditional Statements & Loops

Day 6 - Conditional Expressions

Topics

- if
- if-else
- nested if
- ternary operator

Programs

- Positive or Negative
- Largest of two numbers
- Largest of three numbers

Challenge

Check leap year.

Day 7 - Switch Statement

Programs

- Menu calculator
- Week day
- Month name
- Simple ATM menu

Challenge

Create menu driven calculator.

Day 8 - Conditional Programs

Programs

- Grade calculation
- Voting eligibility
- Even/Odd
- Divisible by 5 and 11
- Character is vowel or consonant

Challenge : Electricity bill calculator.

Day 9 - Loops

Topics

- while
- do while
- for

Difference

```
while
Checks first

do while
Executes once then checks
```

Programs

- Print 1 to 100
- Reverse counting
- Sum of first N numbers

Challenge

Multiplication table.

Day 10 - Loop Programs

Programs

- Factorial
- Fibonacci
- Prime Number
- Palindrome
- Armstrong Number

Challenge

Perfect Number.

Day 11 - Nested Loops

Programs

Print

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

Print

```
1
12
123
1234
```

Challenge : Pyramid patterns.

Week 2 – Conditional Statements & Loops

Days 6–11 Learning Material

Week Learning Objectives

By the end of Week 2, students will be able to:

- Make decisions using `if`, `if-else`, nested `if`, and `switch`.
- Write menu-driven programs.
- Use `while`, `do-while`, and `for` loops.
- Solve mathematical programming problems.
- Create pattern programs using nested loops.

Day 6 – Conditional Expressions (Decision Making)

Learning Objectives

After completing this lesson, students will be able to:

- Understand decision-making in C.
- Use `if`, `if-else`, nested `if`, and the ternary operator.
- Write programs that make decisions based on user input.

What is a Conditional Statement?

A conditional statement allows a program to **make decisions** based on whether a condition is **true** or **false**.

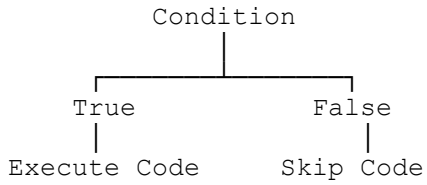
Real-Life Example

```
If it is raining,
    Take an umbrella.
Else
    Don't take an umbrella.
```

Similarly, in C:

```
If marks >= 35
    Pass
Else
    Fail
```

Flow of an if Statement



1. The if Statement

Syntax

```
if(condition)
{
    // statements
}
```

Example

```
#include<stdio.h>

int main()
{
    int age = 20;

    if(age >= 18)
    {
        printf("Eligible to Vote");
    }

    return 0;
}
```

2. The if-else Statement

Syntax

```
if(condition)
{
    // true block
}
else
{
    // false block
}
```

Example

```
#include<stdio.h>

int main()
{
    int age;

    printf("Enter Age: ");
    scanf("%d",&age);

    if(age >= 18)
        printf("Eligible");
    else
        printf("Not Eligible");

    return 0;
}
```

3. Nested `if`

A nested `if` means **an `if` statement inside another `if` statement**.

Example

```
if(age >=18)
{
    if(age<=60)
    {
        printf("Eligible");
    }
}
```

---

4. Ternary Operator (?:)

The ternary operator is a short form of `if-else`.

Syntax

```
condition ? statement1 : statement2;
```

Example

```
#include<stdio.h>

int main()
{
    int a=10,b=20;

    (a>b)?printf("A is Greater"):printf("B is Greater");

    return 0;
}
```

Programs

Program 1 – Positive or Negative Number

```
#include<stdio.h>

int main()
{
    int num;

    printf("Enter Number: ");
    scanf("%d",&num);

    if(num>=0)
        printf("Positive");
    else
        printf("Negative");

    return 0;
}
```

Program 2 – Largest of Two Numbers

```
#include<stdio.h>

int main()
{
    int a,b;

    scanf("%d%d",&a,&b);

    if(a>b)
        printf("%d is Greater",a);
    else
        printf("%d is Greater",b);

    return 0;
}
```

Program 3 – Largest of Three Numbers

```
#include<stdio.h>

int main()
{
    int a,b,c;

    scanf("%d%d%d",&a,&b,&c);

    if(a>b && a>c)
        printf("%d is Largest",a);
    else if(b>c)
        printf("%d is Largest",b);
    else
        printf("%d is Largest",c);

    return 0;
}
```

Challenge

Check Leap Year

Conditions:

- Divisible by 400 → Leap Year
- Divisible by 4 but not by 100 → Leap Year
- Otherwise → Not a Leap Year

Practice Programs

- Positive or Negative
- Even or Odd
- Largest of Two Numbers
- Largest of Three Numbers
- Smallest of Three Numbers
- Pass or Fail
- Leap Year

Day 7 – Switch Statement

What is `switch`?

The `switch` statement is used when a program has **multiple choices**.
Instead of writing many `if-else` statements, we can use `switch`.

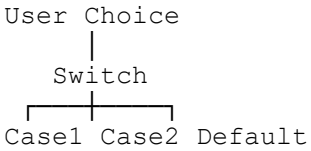
Syntax

```
switch(choice)
{
    case 1:
        statements;
        break;

    case 2:
        statements;
        break;

    default:
        statements;
}
```

Flow



Program 1 – Menu Calculator

```
#include<stdio.h>

int main()
{
    int choice,a,b;

    printf("1.Add\n2.Subtract\n");
    scanf("%d",&choice);

    printf("Enter Two Numbers: ");
    scanf("%d%d",&a,&b);

    switch(choice)
    {
        case 1:
            printf("%d",a+b);
            break;

        case 2:
            printf("%d",a-b);
            break;

        default:
            printf("Invalid Choice");
    }

    return 0;
}
```

Program 2 – Week Day

Input:

```
1
```

Output

```
Monday
```

Program 3 – Month Name

Input

```
5
```

Output

```
May
```

Program 4 – Simple ATM Menu

```
1. Balance
2. Deposit
3. Withdraw
4. Exit
```

Challenge

Create a Menu-Driven Calculator with:

- Addition
- Subtraction
- Multiplication
- Division
- Modulus

Day 8 – Conditional Programs

Programs

Grade Calculation

```
90-100 → A

80-89 → B

70-79 → C

Below 35 → Fail
```

Voting Eligibility

```
Age >=18

Eligible

Else

Not Eligible
```

Even or Odd

Use `%` operator.

Divisible by 5 and 11

```
if(number%5==0 && number%11==0)
```

Character is Vowel or Consonant

Check

```
A,E,I,O,U

a,e,i,o,u
```

Challenge : Electricity Bill Calculator

Example

```
0-100 units

₹2/unit

101-200

₹3/unit

Above 200

₹5/unit
```

Day 9 – Loops

Why Loops?

Suppose you want to print

```
Hello
```

100 times.

Without loop

```
printf();

printf();

printf();

...
```

With loop

```
for(i=1;i<=100;i++)
```

One statement prints it 100 times.

Types of Loops

- while
- do-while
- for

---

while Loop

Checks the condition first.

Syntax

```
while(condition)
{
    statements;
}
```

---

do-while Loop

Executes once, then checks the condition.

Syntax

```
do
{
    statements;

}while(condition);
```

---

for Loop

Best when the number of repetitions is known.

```
for(initialization;condition;increment)
{
    statements;
}
```

Difference

while	do-while
Checks first	Executes first
May execute 0 times	Executes at least once

Programs

Print 1–100

```
for(i=1;i<=100;i++)
```

Reverse Counting

```
100

99

98
```

Sum of First N Numbers

Formula

```
sum=sum+i;
```

Challenge

Print Multiplication Table

```
5 x 1 = 5

5 x 2 =10

...

5 x10=50
```

Day 10 – Loop Programs

Program 1

Factorial

```
5!

5×4×3×2×1

120
```

Program 2

Fibonacci Series

```
0

1

1

2

3

5

8

13
```

Program 3

Prime Number

Check divisibility from 2 to √n (or from 2 to n-1 for beginners).

Program 4

Palindrome Number

```
121
```

```
Reverse
```

```
121
```

```
Palindrome
```

Program 5

Armstrong Number

```
153
```

```
1³+5³+3³
```

```
153
```

Challenge

Perfect Number

Example

```
6
```

```
Factors
```

```
1
```

```
2
```

```
3
```

```
Sum
```

```
6
```

Perfect Number

Day 11 – Nested Loops

What is a Nested Loop?

A loop inside another loop.

Syntax

```
for(i=1;i<=5;i++)
{
    for(j=1;j<=5;j++)
    {

    }
}
```

Pattern 1

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

Pattern 2

```
1
12
123
1234
```

Pattern 3

```
A
AB
ABC
ABCD
```

Pattern 4

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

Challenge

Pyramid Pattern

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

Week 2 Practice Set (30 Programs)

Day 6 (7 Programs)

1. Positive or Negative
2. Even or Odd
3. Largest of Two Numbers
4. Largest of Three Numbers
5. Smallest of Three Numbers
6. Pass or Fail
7. Leap Year

Day 7 (5 Programs)

8. Menu Calculator
9. Week Day
10. Month Name
11. ATM Menu
12. Simple Restaurant Menu

Day 8 (5 Programs)

13. Grade Calculator
14. Voting Eligibility
15. Divisible by 5 and 11
16. Vowel or Consonant
17. Electricity Bill

Day 9 (5 Programs)

18. Print 1–100
19. Reverse Counting
20. Sum of First N Numbers
21. Multiplication Table
22. Print Even Numbers

Day 10 (5 Programs)

23. Factorial
24. Fibonacci
25. Prime Number
26. Palindrome Number
27. Armstrong Number

Day 11 (3 Programs + Challenge)

28. Star Triangle
29. Number Triangle
30. Reverse Star Pattern

Bonus Challenge: Centered Pyramid Pattern

Week 2 Learning Outcomes

After completing Week 2, you will confidently:

- ✅ Make decisions using `if`, `if-else`, nested `if`, `else-if`, and the ternary operator.
- ✅ Develop menu-driven applications using `switch`.
- ✅ Solve real-world conditional programming problems.
- ✅ Master `while`, `do-while`, and `for` loops.
- ✅ Implement mathematical algorithms such as factorial, Fibonacci, prime, palindrome, Armstrong, and perfect numbers.
- ✅ Create star, number, and alphabet patterns using nested loops.
- ✅ Be well prepared for lab exams, technical interviews, and advanced C programming topics in Week 3 (Arrays, Strings, Functions, and Pointers).