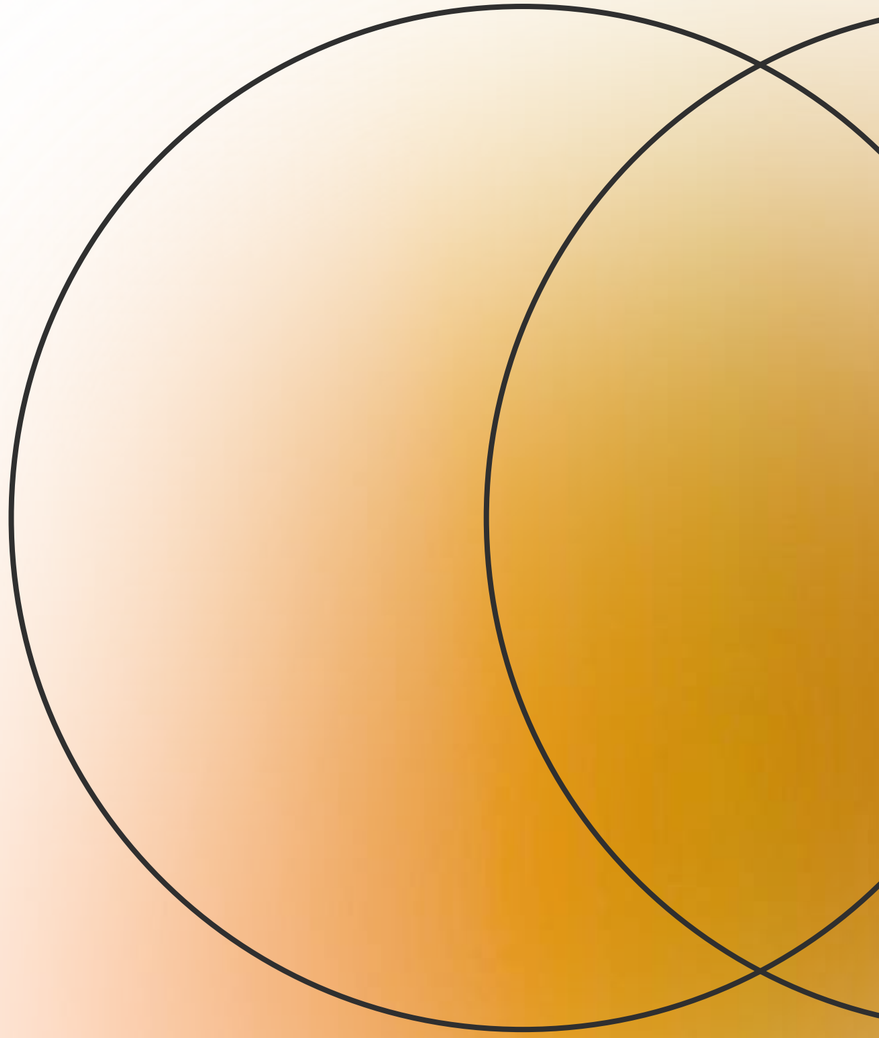
Programming

Fundamentals

Table of Contents

SECTION 1:

FOR Loop



FOR Loop basic Concept

- **Purpose:** Used when number of iterations is known.

Syntax:

```
for(initialization; condition; update) {
        // body of loop
}
```

- Flow:
  Initialize variable → Check condition →
  Execute body → Update → repeat if true.

EXAMPLE

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 5; i++) {
        printf("%d ", i);
    }
    return 0;
}
```

OUTPUT:

Q1

1. **Find the sum of digits of a given number using a for loop.**

Hint: num %10 = last digit of the number

Q2

2. Find the factorial of a number using a for loop.

Hint: 5!= 5*4*3*2*1

SECTION 2:

NESTED FOR Loop

Nested FOR Loop

- Purpose: Loop inside another loop (useful for patterns, matrices).
- Syntax:

```
for(i = 1; i <= n; i++) {
  for(j = 1; j <= m; j++) {
    // inner loop statements
  }
}
```

EXAMPLE

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 3; i++) {
        for(int j = 1; j <= 3; j++) {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

Q1

Q1. Print pattern:

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

Q2

Q1. Print pattern:

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

Q3: Print all factor pairs of a number
Example input: n = 12
Output: (1, 12) (2, 6) (3, 4) (4, 3) (6, 2) (12, 1)

Q4: Display pairs whose sum is even (1–n range)
Example input: n=5
Output: (1,1) (1,3) (1,5) (2,2) (2,4) (3,1) (3,3) ...

SECTION 3:

WHILE Loop

WHILE concept

Purpose: Used when number of iterations is not known in advance.

Syntax:

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

Flow: Check condition first → execute body if true → repeat.

EXAMPLE

```
#include <stdio.h>
int main() {
    int i = 1;
    while(i <= 5) {
        printf("%d ", i);
        i++;
    }
    return 0;
}
```

OUTPUT:

Practice

- **Q1: Print all odd numbers between 1 and 20.**
- **Q2: Reverse a number (e.g., 1234 → 4321) using while loop.**

SECTION 3:

Nested WHILE Loop

Nested WHILE concept

Purpose: Repeat one while loop inside another

Syntax:

```
while(condition1) {
   while(condition2) {
      // inner loop
   }
}
```

Practice

- **Q1: Print the pattern:**

A
B B
C C C
D D D D
E E E E E

- **Q2: Multiplication Table (1 to 5 tables)**

SECTION 3:

DO WHILE Loop

DO WHILE concept

Purpose: Executes at least once even if condition is false.
Syntax:

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

Practice

- **Q1. Take numbers until user enters 0.**
- **Q2. Sum of digits of a number**

SECTION 3:

Nested DO WHILE Loop

Nested DO WHILE concept

Purpose: Use when inner loop depends on outer loop but must run at least once.

Syntax:

```
do {
   do {
      // inner loop body
   } while(condition2);
} while(condition1);
```

Practice

Q1. Find all 3-digit “special numbers” where sum of digits = 9

Thank You