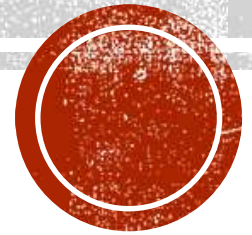


PROGRAMMING LANGUAGES 1

Loops



for LOOP

- Repeats a statement or a block **for** a certain number of times using a counter

```
// Print numbers from 1 to 10
for (int i = 1; i <= 10; i++)
    cout << i << endl;
```

Initialization: `int i = 1` ← Executed only once at start (initialize counter to 1)

Looping condition: `i <= 10` ← Loop exits when condition is false

Increment counter: `i++` ← Executed after each iteration

EXECUTION

- Initially (Step 1): $i = 1 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 1$
- Step 2 ($i++$): $i = 2 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 2$
- Step 3 ($i++$): $i = 3 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 3$
- Step 4 ($i++$): $i = 4 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 4$
- Step 5 ($i++$): $i = 5 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 5$
- Step 6 ($i++$): $i = 6 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 6$
- Step 7 ($i++$): $i = 7 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 7$
- Step 8 ($i++$): $i = 8 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 8$
- Step 9 ($i++$): $i = 9 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 8$
- Step 10 ($i++$): $i = 10 \rightarrow i \leq 10 ? \text{True} \rightarrow \text{Print } 10$
- Step 11 ($i++$): $i = 11 \rightarrow i \leq 10 ? \text{False} \rightarrow \text{Exit loop}$

while LOOP

- Repeats **while** condition is true
- Number of iterations is not known in advance

```
while (condition)  
    statement or block;
```

while LOOP - EXAMPLE

Add numbers until sum exceeds 100

```
int x, sum = 0;
while (sum < 100)
{
    cin >> x;
    sum += x;
}
```

do-while LOOP

- Similar to while loop but iterates at least once

do

statement or block;

while (condition) ; **← semicolon**

do-while LOOP - EXAMPLE

Add numbers until user enters 0

```
int x, sum = 0;
do
{
    cin >> x;    // Read input at least once
    sum += x;
}
while (x != 0);
```

break

- Break loop in the middle of iterations
- Example: Add numbers until sum exceeds 100 & stop if user enters 0

```
int x, sum = 0;
while (sum < 100)
{
    cin >> x;
    if(x == 0) break;
    sum += x;
}
```



continue

- Continue to next iteration of the loop
- Skips the rest of the current iteration
- Example: Add positive numbers until sum exceeds 100

```
int x, sum = 0;
while (sum < 100)
{
    cin >> x;
    if(x < 0) continue;
    sum += x;
}
```



while(true)

- Infinite loop!!
- Stopping condition is not known in advance
- Example: Add numbers until user enters 0

```
int x, sum = 0;
while(true)
{
    cin >> x;
    if(x == 0) break;
    sum += x;
}
```

SUMMARY

- for Loop
- while Loop
- do-while Loop
- break
- continue