Control Statements

Subject: C Language   Date: 30 / 12 / 2025

I. Iterative Control Statements :

i) while (Entry Control Loop)
ii) do while (Exit Control Loop)
iii) for (Entry Control Loop)

- while loop :- The while loop is used when you don't ~~know~~ necessarily know how many times the loop will run, but you know the condition that should stop it. It checks the condition before executing the code.

  - Best for : Reading data until the end of a file, waiting for a specific user input, or any scenario where the "end" depends on dynamic logic.

  - Risk : If the condition is never false, you get an "infinite loop" that can crash your program.

Example :-

```
initialization;
while (condition) {
    statement;
    increment/Decrement;
}
```

Subject: ______ Date: __/__/__

do while loop :— The do-while loop is a variation of the while loop. The key difference is that it checks the condition after the code block has executed. This guarantees that the code inside the loop will run at least once.

- Best for : Menus (where you want to show the options at least once before checking if the user wants to quit) or input validation.

Example :—

```
initialization;
do{
    statement;
    increment/decrement;
} while(condition);
```

for loop :— The for loop is generally used when you know exactly how many times you want to run a block of code. It groups the initialization, the condition, and the increment/decrement all in one line.

- Best for : Counting, iterating through arrays, or running code a fixed number of times.

Subject: Date: / /

Example :-

```
for(initialization ; Condition ; increment/decrement)
{
    Statements ;
}
```

NOTE :- for loop ke andar agar Condition nahi bhi likhte hai to wah true maan liya Jata hai.

2. Break keyword :-

```
while(condition)
{
    ________
    ________
    ________
    break;
    ________
}
```

NOTE :-
- Break is a keyword.
- It can be used in the body of loop or in the body of Switch.
- when break encounters loop terminates and Control move out of the loop body.

3. Continue keyword :-

```
while (condition)
{
    ______
    ______

    ______
    Continue;
    ______
    ______
}
```

NOTE:-
- Continue is a keyword.
- Continue can only be used in the body of loop.
- Continue transfers the control to the next iteration.

4. Nested loop :-

⇒ when we write loop inside another loop is called nested loop.

Example :-

```
while (condition)
{
    while (condition)
    {
        ______
        ______
    }
}
```

Subject: ______ Date: __/__/__

5. Difference between while, for and do-while loop.

⇒

Features	while loop	for loop	do-while loop
Type	Entry-Controlled loop	Entry-Controlled loop	Exit-Controlled loop
Condition check	Before loop body	Before loop body	After loop body
Minimum execution	0 times	0 times	At least 1 time:
Initialization	Done before loop	Done inside loop	Done before loop.
Condition	written in while	written in for	written in while
Increment/Decrement	Inside body loop	Inside loop header	Inside loop body
Semicolon usage	No Semicolon	No Semicolon	Semicolon required
Best used when	Iteration unknown	Iteration known	Loop must run once