

Answer the following

1. State the difference between while loop, for loop and do while loop.

Ans –

Aspect	while Loop	for Loop	do-while Loop
Definition	A control flow statement that repeatedly executes a block of code as long as the condition is true.	A loop structure designed to repeat a block of code a specified number of times or until a condition is met.	Similar to while, but the code block is executed at least once before checking the condition.
Execution Flow	Checks the condition before entering the loop body.	Checks the condition before entering the loop body.	Executes the loop body once before checking the condition.
Use Case	Ideal for loops where the number of iterations isn't known in advance but depends on a condition.	Best suited for loops with a known number of iterations or when an iterator variable is used.	Used when the loop body must execute at least once, regardless of the condition.
Syntax	<code>while(condition) { // Code block }</code>	<code>for(initialization; condition; increment/decrement) { // Code block }</code>	<code>do { // Code block } while(condition);</code>
Entry/Exit Check	Entry-controlled: The loop might not execute if the condition is initially false.	Entry-controlled: The loop might not execute if the condition is initially false.	Exit-controlled: The loop executes at least once because the condition is checked after the execution of the loop body.
Example	<code>int i = 0; while(i < 5) { printf("%d ", i); i++; }</code>	<code>for(int i = 0; i < 5; i++) { printf("%d ", i); }</code>	<code>int i = 0; do { printf("%d ", i); i++; } while(i < 5);</code>

2. What is the difference between break and continue.

Ans –

Aspect	break	continue
Definition	Terminates the current loop or switch statement and exits out of it immediately.	Skips the remaining statements in the current iteration of the loop and proceeds to the next iteration.
Effect on Loop	Exits the loop entirely, regardless of the remaining iterations.	Does not terminate the loop; it just skips the current iteration and continues with the next one.
Usage	Used to terminate a loop prematurely, often based on a specific condition (e.g., breaking out of infinite or nested loops).	Used to skip processing for certain conditions without exiting the loop.
Execution Flow	Ends loop execution and moves the program control to the statement immediately after the loop.	Moves control directly to the loop's increment/decrement statement (in for) or the condition check (in while and do-while) for the next iteration.
Example in Loop	<pre>for(int i = 0; i < 10; i++) { if(i == 5) break; printf("%d ", i); } //</pre> <p>Output: 0 1 2 3 4</p>	<pre>for(int i = 0; i < 10; i++) { if(i == 5) continue; printf("%d ", i); } //</pre> <p>Output: 0 1 2 3 4 6 7 8 9</p>