

Freebooks.pk

Free Notes • Past Papers • Guides for Students

2nd Year Computer Science — Punjab Board

CHAPTER 12

Loop Constructs

Key Concepts • Definitions • Short & Long Questions • MCQs • Diagrams • Exam Tips

© freebooks.pk — Original educational content by the Freebooks.pk Editorial Team



This chapter covers Loop Constructs from the 2nd Year (ICS Part-II) Computer Science syllabus of the Punjab Curriculum and Textbook Board (PTB/PCTB). It explains iteration and the three loop statements of C, the while loop, the do-while loop and the for loop, together with nested loops, sentinel-controlled loops and type casting. These notes are prepared by freebooks.pk.

Many problems require a statement or set of statements to be executed repeatedly. A loop is a control structure that repeats statements, either a fixed number of times or until a condition is met.

Learning Objectives

- Explain iteration (looping) as the third control structure.
- Use the while loop.
- Use the do-while loop and know why it runs at least once.
- Use the for loop with its three expressions.
- Explain nested loops.
- Explain sentinel-controlled loops and type casting.

Key Concepts

Iteration and the Loop Control Variable

Iteration is the third type of program control structure, after sequence and selection, and the repetition of statements in a program is called a loop. C provides three loop statements: while, do-while and for. Each loop is controlled by a condition and usually by a loop control variable, which is a variable whose value controls the number of iterations. The group of statements repeated by the loop, enclosed in braces, is called the body of the loop.

The while Loop

The while loop keeps repeating its associated statements as long as a specified condition remains true; it is useful when the programmer does not know in advance how many times the loop will run. Its syntax is while (condition) followed by the body. When the condition is true the body is executed, and as soon as the condition becomes false the loop terminates. In a while loop the loop control variable is initialised outside the loop and is incremented or decremented inside the loop body.

The do-while Loop

The do-while loop is similar to the while loop except that the condition is tested at the end of the loop body, which guarantees that the body is executed at least once. Its syntax begins with do, then the body, and ends with while (condition); note that the do-while statement ends with a semicolon, and omitting it causes a syntax error. This loop is useful when statements must run at least once, for example when reading and validating user input until a valid value is entered.



while vs do-while

The key difference is when the condition is tested. In a while loop the condition is tested first, so the body may execute zero times if the condition is false at the start. In a do-while loop the body is executed first and the condition is tested afterwards, so the body always executes at least once. Therefore a while loop is used when the body should run only if the condition is true, while a do-while loop is used when the body must run at least once.

The for Loop

The for loop is the most flexible loop and is preferred by most programmers. Its syntax is for (initialization; test condition; increment/decrement) followed by the body. The three expressions are separated by semicolons: the initialization sets the loop control variable and runs only in the first iteration; the test condition is checked before each pass; and the increment/decrement changes the loop control variable after each pass. The initialization and the increment/decrement are optional, but the test condition is mandatory and cannot be omitted.

Nested Loops

A nested loop is a loop placed inside the body of another loop, and nesting can be done to any level, though complexity increases with each level. There is no restriction on the types of loops that may be nested, so a while, do-while or for loop can be placed inside any other loop. In a nested loop the inner loop completes all its iterations for each single iteration of the outer loop, and each time the outer loop starts a new iteration, the variables used in the inner loop are re-initialised and re-processed.

Sentinel-Controlled Loops

Often a program must read a list of items whose length is not known in advance, for example the marks of every student in a class. A sentinel-controlled loop handles this by asking the user to enter a special value, called a sentinel value, after the last data item. The loop tests each item and exits when the sentinel is read. The sentinel value must be chosen carefully so that it can never occur as normal data; for example, a negative number can be a sentinel for marks because marks are never negative.

Type Casting

Type casting is temporarily treating a variable of one type as another type for a single calculation. It is often needed with division: when two integers are divided the result is an integer and the fractional part is truncated, giving an inaccurate answer. Writing a type in parentheses before a variable, as in (float)total_students, causes that integer variable to act as a float for that one calculation, preserving the fractional part, while it continues to behave as an integer everywhere else.



Important Definitions

Term	Definition
Iteration (loop)	The repetition of a statement or group of statements in a program.
Loop control variable	A variable whose value controls the number of iterations.
Body of the loop	The statements repeated by the loop, enclosed in braces.
while loop	A loop that tests the condition before executing the body.
do-while loop	A loop that executes the body first, then tests the condition (runs at least once).
for loop	A loop with initialization, test condition and increment/decrement in one statement.
Nested loop	A loop placed inside the body of another loop.
Sentinel value	A special value entered to signal the end of input data.

Key Facts & Rules

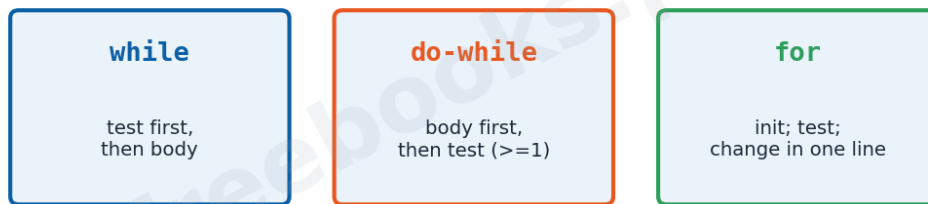
Item	Detail
while	while (condition) { body }
do-while	do { body } while (condition); (ends with ;)
for	for (init; test; change) { body }
for optional parts	init and change optional; test is mandatory
Loop control variable	initialised, tested and changed to control iterations
Sentinel loop	read until a special value (e.g. a negative number)
Type cast	(float)total_students keeps the fractional part

Diagrams & Illustrations

Three loop constructs: the three loop constructs of C, while, do-while and for, and how each tests its condition.



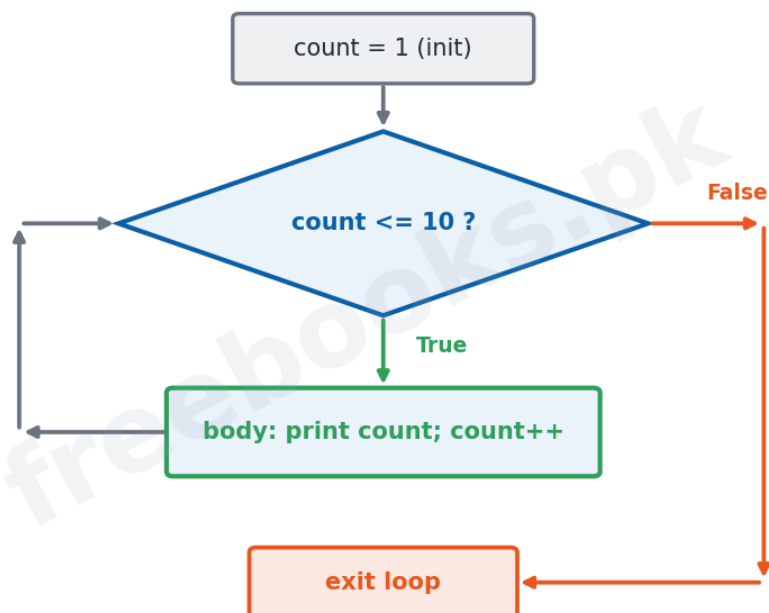
Three Loop Constructs in C



Three loop constructs

Flowchart of the while loop: how a while loop initialises, tests the condition, executes the body and exits when false.

Flowchart of the while Loop



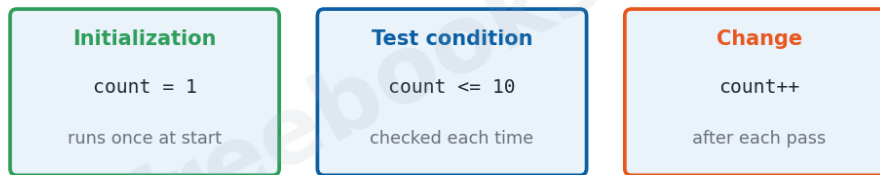
Flowchart of the while loop

The three expressions of a for loop: the initialization, test condition and increment/decrement parts of a for loop.



The Three Expressions of a for Loop

```
for ( count = 1 ; count <= 10 ; count++ )
```



The three expressions of a for loop

Solved Examples & Practice

Print 1 to 10 (while)

Initialise count = 1, then while (count <= 10) print count and do count++; the loop prints 1 to 10.

Validate input (do-while)

do { read state } while (state != 'w' && state != 'd'); forces the user to enter a valid value.

Print 1 to 10 (for)

for (count = 1; count <= 10; count++) printf("%d", count); prints 1 to 10 in one compact statement.

Average with sentinel

Read marks in a while loop until a negative number is entered, summing them, then divide using (float) to get the average.

Short Questions & Answers

Q1. What is a loop?

Ans. A control structure that repeats a statement or group of statements in a program.

Q2. What is a loop control variable?

Ans. A variable whose value controls how many times the loop runs.

Q3. Why does a do-while loop always run at least once?

Ans. Because its condition is tested at the end, after the body has already executed once.

Q4. Name the three expressions of a for loop.

Ans. Initialization, test condition and increment/decrement.



Q5. What is a nested loop?

Ans. A loop placed inside the body of another loop.

Q6. What is a sentinel value?

Ans. A special value entered after the last data item to signal the loop to stop.

Long Questions & Answers

Q1: Explain the while loop and the do-while loop, and state the difference between them.

Iteration, the repetition of statements, is the third program control structure after sequence and selection, and C provides three loop statements to carry it out. The while loop keeps repeating its body as long as a specified condition remains true, and its syntax is the keyword while followed by a condition in parentheses and then the body of the loop. It is especially useful when the programmer does not know in advance how many times the loop will run. When execution reaches the while statement the condition is tested; if it is true the body is executed and control returns to test the condition again, and as soon as the condition becomes false the loop terminates and control passes to the statement after the loop. In a while loop the loop control variable, the variable whose value controls the number of iterations, is initialised outside the loop and is incremented or decremented inside the body, for example a variable count initialised to 1 and increased by one each time until the condition `count <= 10` becomes false. The do-while loop is very similar but tests its condition at the end of the body rather than at the beginning; its syntax begins with the keyword do, then the body, and ends with while (condition) followed by a semicolon, and forgetting this semicolon causes a syntax error. Because the condition is tested after the body, a do-while loop always executes its body at least once, which makes it ideal for tasks such as reading and validating input, where the program must obtain a value at least once and keep asking until the value is valid. The essential difference between the two is therefore the timing of the test: in a while loop the condition is tested first, so the body may run zero times, whereas in a do-while loop the body runs first and is guaranteed to execute at least once.

Q2: Explain the for loop with its three expressions and describe nested loops.

The for loop is another way of writing a loop in C and, because of its flexibility, it is preferred by most programmers for counting loops. Its syntax is the keyword for followed by three expressions in parentheses, separated by semicolons, and then the body: `for (initialization expression; test condition; increment/decrement expression)`. The initialization expression sets the starting value of the loop control variable and is executed only once, during the first iteration. The test condition is then checked; if it is true the body of the loop is executed, after which the increment/decrement expression is evaluated to change the loop control variable, and the condition is tested again. This cycle of test, execute body, and change continues as long as the condition is true, and when the condition becomes false the loop terminates and control passes to the next statement. For example, `for (count = 1; count <= 10; count++)` prints the numbers 1 to 10. Two of the three expressions, the initialization and



the increment/decrement, are optional and may be omitted, but the test condition is mandatory; if the initialization is omitted the loop control variable must be initialised before the loop and changed inside the body. A nested loop is a loop placed inside the body of another loop, and nesting may be carried to any level, although the complexity grows with each level. Any type of loop may be nested inside any other, so a while, do-while or for loop can appear inside a for loop and so on. In a nested loop the inner loop runs through all its iterations for every single iteration of the outer loop, and each time the outer loop begins a new iteration the variables of the inner loop are re-initialised. Nested loops are commonly used to produce two-dimensional patterns, such as printing rows of asterisks, where the outer loop controls the rows and the inner loop controls the items in each row.

Q3: What is a sentinel-controlled loop? Explain it with the idea of type casting.

Many programs need to read and process a list of items whose length is not known in advance; for instance, to find the average marks of a class the program must read the marks of every student, but the number of students may vary. A sentinel-controlled loop solves this problem by asking the user to enter a special value, called a sentinel value, after the last real data item. The loop reads and processes each item, and the loop condition tests every value so that the loop exits as soon as the sentinel value is read. The sentinel value must be chosen carefully so that it can never occur as valid data: for reading marks, a negative number is a good sentinel because a student can never score negative marks, whereas zero would be a poor choice since a student may score zero. In a typical average program the loop keeps reading marks and adding them to a running sum while counting the students, and stops when a negative number is entered. Before dividing to find the average, the program should check that at least one student was entered, because dividing by a total of zero would cause a runtime division-by-zero error. The average is then found by dividing the sum by the number of students, and here type casting becomes important. Since both the sum and the count are integer variables, their division would be an integer division in which the fractional part is truncated, giving an inaccurate result. To avoid this, the type float is written in parentheses before the count, as in `(float)total_students`, which causes that integer variable to act temporarily as a float for that one calculation so that the fractional part of the answer is preserved; the variable continues to behave as an integer in all other calculations. This temporary change of type for a single calculation is called type casting.

MCQs with Answers

1. The repetition of statements in a program is called a:

(a) branch (b) loop (c) function (d) header

Correct Answer: (b) loop. Loop.

2. The three loop statements in C are while, do-while and:

(a) switch (b) for (c) if (d) case

Correct Answer: (b) for. for.

3. The loop that tests its condition at the end is the:



(a) while (b) for (c) do-while (d) nested

Correct Answer: (c) do-while. do-while loop.

4. A do-while loop executes its body at least:

(a) zero times (b) once (c) twice (d) ten times

Correct Answer: (b) once. Once.

5. The do-while statement ends with a:

(a) colon (b) semicolon (c) comma (d) brace

Correct Answer: (b) semicolon. Semicolon.

6. In a for loop, the part that runs only once is the:

(a) test condition (b) initialization (c) increment (d) body

Correct Answer: (b) initialization. Initialization.

7. Which expression of a for loop is mandatory?

(a) initialization (b) test condition (c) increment (d) none

Correct Answer: (b) test condition. Test condition.

8. A loop inside another loop is called a:

(a) chained loop (b) nested loop (c) switch (d) sentinel

Correct Answer: (b) nested loop. Nested loop.

9. A special value that signals the end of input is a:

(a) counter (b) sentinel value (c) flag (d) token

Correct Answer: (b) sentinel value. Sentinel value.

10. (float)total_students in an average formula is an example of:

(a) assignment (b) type casting (c) declaration (d) looping

Correct Answer: (b) type casting. Type casting.

Quick Revision Summary

- Iteration (loop) = repetition of statements; C has while, do-while and for loops.
- while tests condition first (may run 0 times); do-while tests at end (runs at least once, ends with ;).
- for (init; test; change) — init runs once, test mandatory, init and change optional.
- Nested loop = loop inside a loop; inner loop completes fully for each outer iteration.
- Sentinel-controlled loop reads until a special value (e.g. a negative number) that can't be real data.
- Type casting: (float)var keeps the fractional part in integer division. Notes by freebooks.pk.

Exam Tips

- Explain iteration and name the three loops.
- Give the syntax of while and do-while and their difference.
- Explain the three expressions of the for loop; note which are optional.
- Define a nested loop and describe how inner/outer loops run.



- Explain a sentinel-controlled loop with a suitable sentinel value.
- Explain type casting with the (float) example and division by zero.

© freebooks.pk — DMCA-protected. May be shared or linked without changes. Original content by the Freebooks.pk Editorial Team.

