9.1 While Repetition control

Content:

While loop

* Loop evaluation
* Break / Continue

Learning Outcomes:

Students should be able to

9.1.1 Define the term pre-test loop and entrance-controlled loop

9.1.2 Define the term fixed-count loop

9.1.3 Define the term variable condition loop

9.1.4 Explain the purpose of the while statement in C/C++

9.1.5 Determine the application flow when a while statement is encountered

9.1.6 Identify or correct the general form for a while statement

9.1.7 Write C/C++ code that will use relational expressions in while statement. (Max 2 levels only)

9.1.8 Write C/C++ code that will use relational expressions in while statements with compounded content (Max 2 two nested levels only)

9.1.9 Explore the concept of block scope when writing while statements with compounded content

9.1.10 Explain the difference between fixed-count loop and variable condition loop

9.1.11 Write C/C++ code that will use relational expressions in while statements with interactive input (Max 2 two nested levels only)

9.1.12 Explain what a break statement is used for

9.1.13 Explain what the continue statement is used for

9.1.14 Write C/C++ code that will use relational expressions in while statements making use of the break statement

9.1.15 Write C/C++ code that will use relational expressions in while statements making use of the continue statement

9.2. Logic operators in While loop

Content:

* While loop
* Logic operators

Learning Outcomes:

Students should be able to

9.2.1 Write C/C++ code that will use relational expression containing logic operators in while (Max 3 level, Max 3 logic operators per level)

9.2.2 Write C/C++ code that will use relational expressions containing logic operator in while loop with compounded content (Max 3 level, Max32 logic operators per level)

9.3. Do While loop

Content:

* Do While loop
* Logic operators

Learning Outcomes:

Students should be able to

9.3.1 Define the term post-test loop and exit controlled loop

9.3.2 Determine the application flow when a do while statement is encountered

9.3.3 Identify or correct the general form for a do while statement

9.3.4 Define the term sentinel

9.3.5 Explain where and for what reason a sentinel is used

9.3.6 Write C/C++ code that will use relational expression containing logic operators in a do while loop (Max31 level, Max 3 logic operators per level)

9.3.7 Write C/C++ code that will use relational expressions containing logic operator in a do while loop with compounded content (Max 3 level, Max 3 logic operators per level)

9.3.8 Write C/C++ code that will use relational expression containing logic operators in a sentinel controlled do while loop (Max 3 level, Max 3 logic operators per level)

9.3.9 Write C/C++ code that will use relational expressions containing logic operator in a sentinel controlled do while loop with compounded content (Max 3 level, Max 3 logic operators per level)

9.4. The for loop

Learning Outcomes:

Students should be able to

Content:

* For loop

Learning Outcomes:

Students should be able to

9.4.1 Define the term variable condition loop

9.4.2 Explain how the, for statement differs from the while

9.4.3 Explain how the, for statement is similar to the while

9.4.4 Identify or correct the general form for a for statement

9.4.5 Explain the purpose of each part in the for-loop initialisation

9.4.6 Explain what effect an empty semicolon will have on the for loop

9.4.7 Explain the term infinite loop

9.4.8 Write C++ code to create an infinite for loop.

9.4.9 Use the break statement to end an infinite for loop based on the loop reaching a condition.

9.4.10 Determine the application flow when for statement is encountered.

9.4.11 Write C++ code that will use a for loop with a predetermined number of loops.

9.4.12 Write C++ code that will use a for loop with a non-sequential counter variable.

9.5. Nested loops

Learning Outcomes:

Students should be able to

Content:

Nested loops

9.5.1 Define the term nested loop

9.5.2 Define the terms inner and outer loops

9.5.3 Write C++ code that will nest identical type loops (Max 3 levels)

9.5.4 Write C++ code that will nest different type loops (Max 3 levels)

9.5.5 Determine the application flow when nested loop statement is encountered

9.6. Program flow and control

Learning Outcomes:

Students should be able to

Content:

* Break and continue statements

9.6.1 Explain what a break statement is used for

9.6.2 Explain what the continue statement is used for.

9.6.3 Implement a break or continue statement as required as part of a code solution.

9.7. Program flow and control

Learning Outcomes:

Students should be able to

Content:

* Creating C/C++ solutions

9.7.1 Plan, Design, create and code a C++ solution incorporating various structures and nested strictest to solve a given problem.