

Summary of Chapter 5: Repetition Structures

This summary will explain repetition structures and their use in programs.

A repetition structure causes a statement or a set of statements to execute repeatedly. Repetition structures are used when instructions are given to the computer without having the need to write many lines for the same task. An example of a simple act such as taking a walk to your destination, where you move one leg after the other until you have reached your destination. This is an example of loops. It is repetition getting closer to the end goal until it is reached.

There are different kinds of loops in programming. There are two broad categories of loops. One is called condition-controlled and count-controlled. A condition-controlled loop uses a boolean (true/false) to control the number of times the loop repeats. A count-controlled loop repeats a specific number of times.

The structure for condition-controlled loops starts with either a “While”, “Do-while” and “Do-until”. The While loop performs a task while something is true/false, or while a condition is true. A While loop is a pretest loop in which it checks the condition before performing a task.

Loops must contain within themselves a way to terminate. Such as a value being modified to satisfy the condition or changing the boolean value to a true or a false. If the loop doesn't get the result it is looking for, it continues forever. This is called an infinite loop. An infinite loop continues to repeat until the the program is interrupted.

Using modules in a loop is a great way to improve the design of the program. Modules being called can perform a task, then new information can be brought back to the loop, and have the loop check if the condition is met with this new information.

Count-controlled loops mostly used with the For statement. For loops is a count-controlled loop that iterates a specific number of times. An example for using a count-controlled loop can be a business that is open six days a week where the amount of sales would need to be inputted everyday for six days. When creating counter-controlled loops, a counter variable is commonly used for either incrementing by a number or decrementing until a specific number is reached that would break out of the loop.

The number that helps the loop terminate is called a Sentinel. A Sentinel is a special value that marks the end of a list of values. The program knows it has reached the end of the list once it reads a sentinel value.

Loops help programs run efficiently and quickly as long as the loop has accurate information it can reference from. Otherwise it can go horribly wrong. Tracing can take very long when it comes to loops so it is very important that your logic is precise before writing your loop.