Computer programming argumentative

In computer science and computer programming, the loop (or iteration) is a series of instructions or programming statements that are executed depending on certain conditions.

The loop is also used to refer to the order in which these instructions (or statements) are executed. Within the programming language, execution of the loop would result in changes in the subsequent control flow of instructions, depending on the conditions by which the loop exists. Loops can increment a counter, and typically works this way: First, a loop checks whether or not a certain condition is reached.

For example, it can check whether or not a counter has reached a certain number (condition). If the condition is not reached, it moves on the next instruction in the loop sequence (which is usually to go back to the start of the sequence). This process repeats itself until the prescribed condition is reached.

If the condition is reached, the next instruction branches out to a new sequence outside the loop. Programming languages use this fundamental idea and implement their own versions of looping structures.

Most languages like Java and C/C++ use a variation or a derivative of the For and While loops. The For loop is based on a counter, while the While loop is based on a Boolean condition. In Wong’s article, looping structures are used by text-based and graphical-based designs differently. For text-based designs, the programming language usually checks for the loop termination using the While loop construct, while for graphical-based loops, the language uses iteration counts (counters).

Based on Vaishnav’s (2006) article, explain the benefits of change control to your company’s management. The importance of service availability in the IT industry cannot be stressed enough—loss in service availability is not only costly, it can adversely affect a company’s reputation. Research shows that problems in the implementation of changes in management systems and processes within an organization are the major causes of system outages. The problems mainly lie on the “ people” factor.

Traditionally, changes are manually carried out, those implementing the changes not fully comprehending the full nature of the changes. This practice can cause errors and results in service outages. Change control ensures that the application of changes is done in automation—changes can be tracked real-time and they can automatically be implemented and executed. Change can also track who did the changes, when was it done, and on what conditions were the changes made. Change control also allows for documentation of these changes.

In the past, IT organizations relied on manual system scans to keep track of changes and see what may have caused an outage.

With change control, this same thing can be done quickly and easily—if an outage does occur, the problem can be detected with ease and accuracy. In an industry where service is expected to be available at all times, change control proves to be an invaluable tool. Aside from the tracking advantages change control provides, it also ensures that changes within a system are correct and are in-line with the organization’s change policy.