

# Introduction to Computer Science

By Mohamed Essam  
mhmd96.essam@gmail.com



TA, Mohamed Essam



# PROGRAM LOGIC

The central task in computer science is the development of new programs. These programs are created through software engineering in a process similar to that of engineering for a building.

**1. The specifications** of the program are laid out to determine what the program is going to do. Then the program is designed to decide how the program will achieve its goals. Only then is the program written.



# PROGRAM LOGIC

**2. At the design stage**, the programmer determines the program logic using an algorithm, which is a series of instructions that define how to solve a problem. This definition may sound very similar to that of a computer program, but there's an important distinction. An algorithm is described "in English" rather than written in a computer programming language. Thus, an algorithm is written to be "human readable," whereas a program is written to be "machine readable." Put another way, an algorithm is the idea behind the solution, while the program is the actual implementation of that solution.

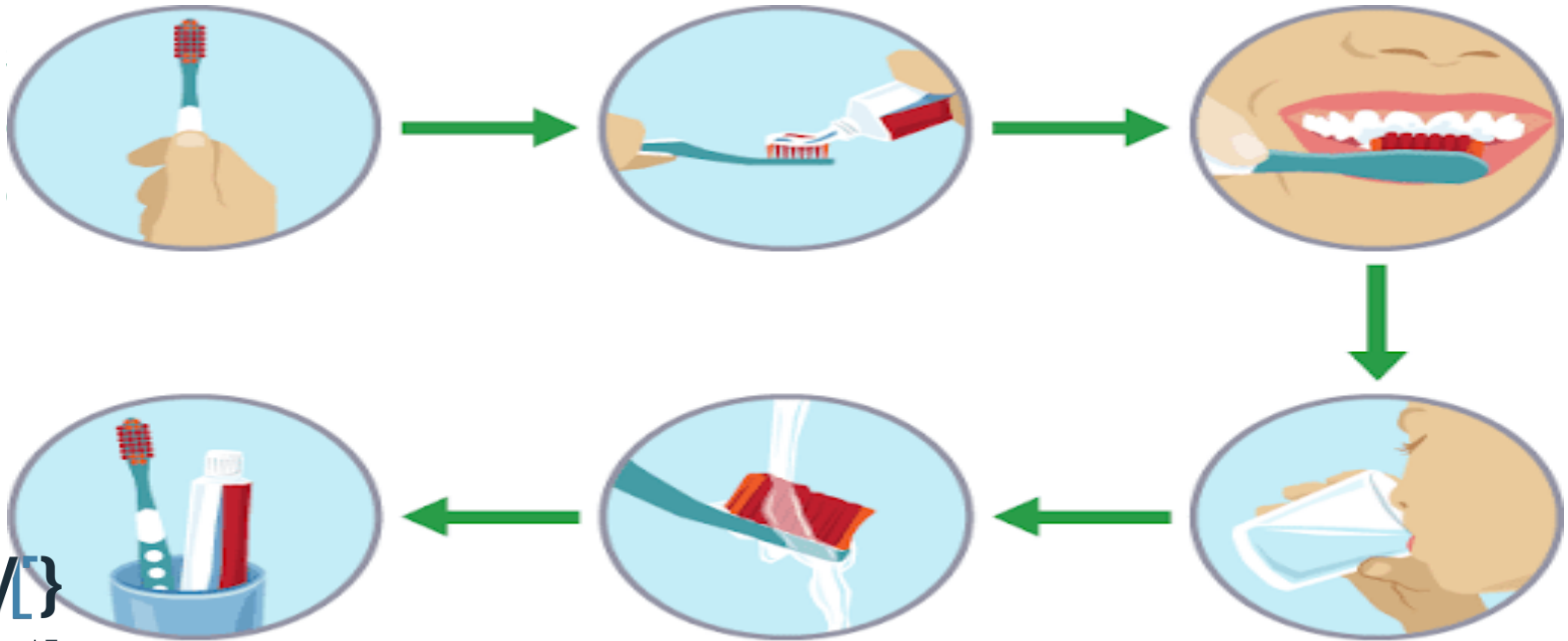


# Sequential Execution

Sequential Execution The sequence is the default control flow; it results from the lack of any control structure and is the easiest to understand. In a computer program, the instructions are executed as they are listed, from top to bottom, unless the computer is told otherwise. Executing instructions in the order listed is sequential execution.



# Sequential Execution



# Conditional Execution

In the previous example, all the instructions have to be executed every time the algorithm is used (every time Todd grooms himself for a party). In other cases an instruction or group of instructions may be optional. With conditional execution, an instruction is executed or skipped based on a condition tested when the instruction is reached. In most algorithms and programming languages, conditional execution can be spotted by looking for the word “if,” as in, “if this condition is true, then do this step.” For that reason, programmers often refer to conditionally executed instructions simply as “if” statements.”



# Conditional Execution



gettyimages®  
Rastko Belic / EyeEm

1069712212

1069712212





# Repetitive Execution


While conditionals are used to execute or skip an instruction, repetitive execution is used when the same instructions are executed more than once. In normal discourse, most programmers refer to repetitive execution as looping, from the idea of circling around to a previous position. Two ways to loop are counterbased and conditional.





# Repetitive Execution

**1.Counting Loop** A counting loop is used when the number of times to repeat the code is known before the loop starts. Todd uses a counterbased loop when he attends parties where he will meet his neurotic, platonic girlfriend, Marta. If he doesn't talk to Marta at least three times, she will become moody and bitter for weeks. After fulfilling his duty with Marta, however, and being in no mood for further conversation, Todd turns up the music volume so that everyone can dance instead of talk.





# Repetitive Execution

**2. Conditional Loop With conditional looping**, a set of instructions is executed until a specified condition is met. Conditional loops do not require knowing how many times the loop will execute. Todd uses a conditional loop at parties when Marta is absent. Freed from any conversational obligations, Todd moves from group to group at the party, settling down with the first group that is not talking about politics.



# SOFTWARE ENGINEERING

Software engineering consists of several phases for the development of the software, phases that can be combined into development paradigms



# SOFTWARE ENGINEERING

## 1. **Specification:**

The first step in any software development project is specification, which determines exactly what abilities the finished software will have. In this phase, one asks the question, “What will the program do?” This step is accomplished in different ways, depending on the destination of the software. Some software is created for use by a particular company or client. For example, a bank might have its own programmers develop new software for calculating mortgage amortization tables. In this case, the programming team can talk directly with the users to see what their needs and desires are.



# SOFTWARE ENGINEERING

**2.Design** In the specification phase, one asks, “What will the program do?” In the next phase, design, one asks, “How will the program do it?” The design phase creates the blueprint for the software’s creation.

In the design phase, the programmers choose a programming language to work in, choose algorithms for different functions of the program, decide which members of the team will do what, make charts describing how the parts of the program will fit together, and so on.



# SOFTWARE ENGINEERING

**3.Implementation In the implementation phase**, the program is actually written. The phase is so-named because the programmers are implementing the design that was made in the previous step. Most people—beginning programmers included—assume that the majority of software development is in implementation, but it's actually just a fraction.



# SOFTWARE ENGINEERING

**4. Testing In the testing phase,** the programming team determines if the software meets the specifications and the design.



# SOFTWARE ENGINEERING

**5.Maintenance** Most programs need support and additional development after they have been released,

which is known as the maintenance phase. Support can include training clients on the software, installing the software on a client's computers, or developing additional product documentation. Additional development is needed when errors are encountered. If an error is extensive, an entirely new version may need to be installed. Small errors can be fixed with a patch, which is a section of code that replaces a part of an already installed program and is usually downloaded from the software developer's Web site.





# LANGUAGES

Computers understand machine language, but programmers use higher-level languages to create programs for the computer. A compiler translates from one language to another.



# Compilation

You might wonder: If machine language is the language the CPU understands, how can one program any other way? One can write in another programming language that is easier to use, then translate the program into the machine language of the CPU used to execute the program. These easier-to-use programming languages are called “high-level languages” because they are at a high level of abstraction from the actual machine hardware. The process of translating a high-level language into a machine language is called compilation.