Computing Science 10

SECTION 7

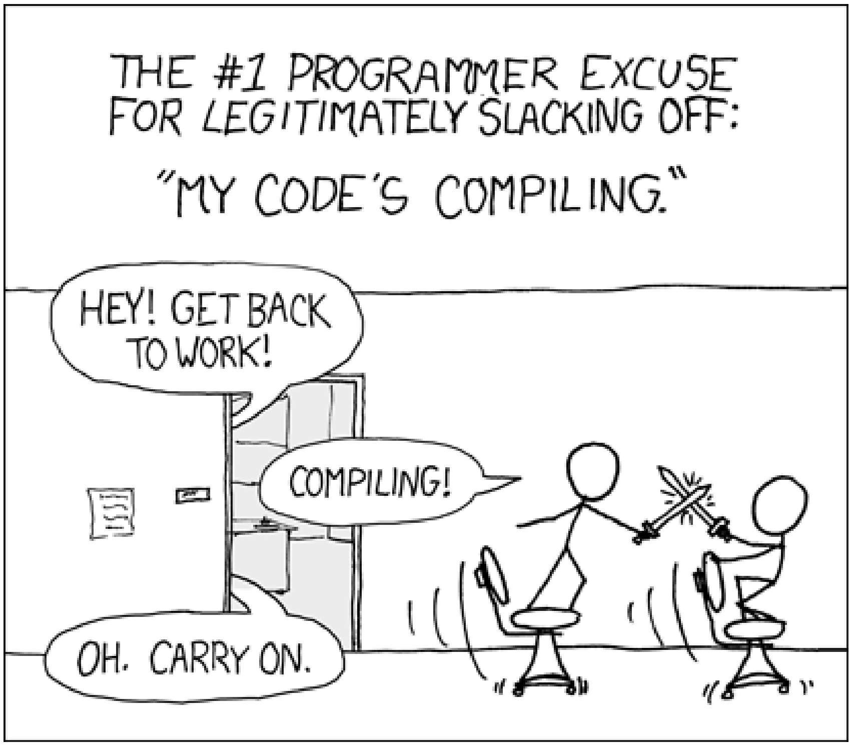
Explain the nature of structured programming

Structured programming is a programming paradigm aimed on improving the clarity, quality, and development time of a computer program by making extensive use of subroutines, block structures and for and while loops.

a subroutine is a sequence of program instructions that perform a specific task, packaged as a unit.

a block is a section of code which is grouped together

Rationale for structured programming



At a low level, structured programs are often composed of simple, hierarchical program flow structures. These are sequence, selection, and iteration.

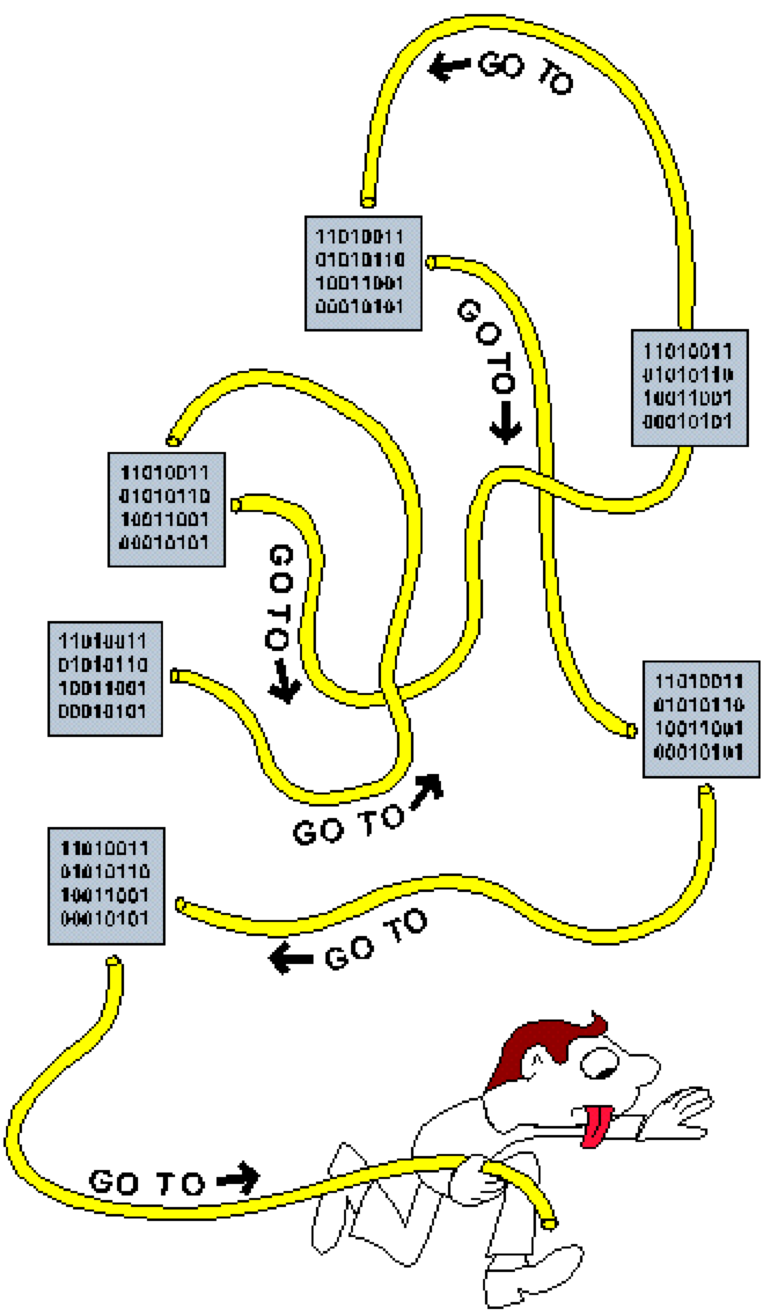
Structured programming is often (but not always) associated with a "top-down" approach to design. It is possible to do structured programming in any programming language.

GOTO-less programming

GOTO is a statement found in many computer programming languages. It performs a one-way transfer of control to another line of code; in contrast a function call normally returns control.

The primary criticism is that code that uses **GOTO** statements is harder to understand than alternative constructions.

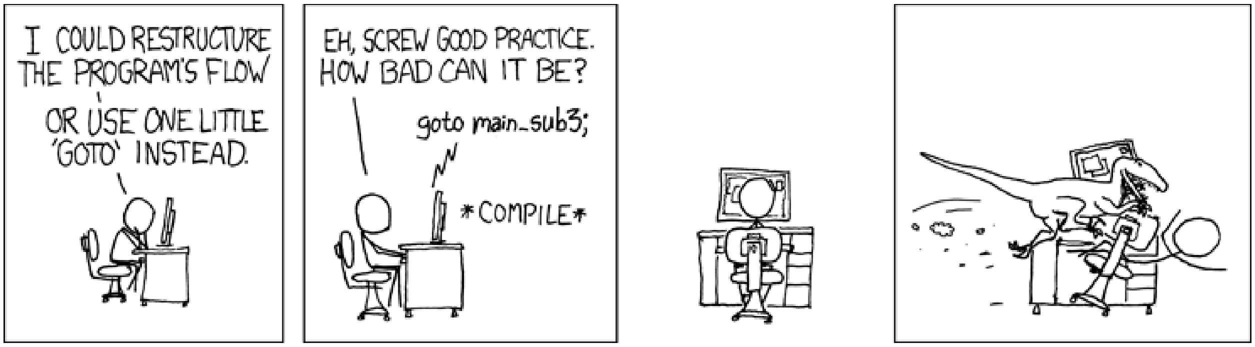
The 1960s and 1970s saw computer scientists move away from **GOTO** statements in favor of the "structured programming" paradigm, with **GOTO** criticized as leading to "unmaintainable spaghetti code".



From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.

Spaghetti code – a derogatory term for source code that has a complex and tangled control structure, especially one using many GOTO's.

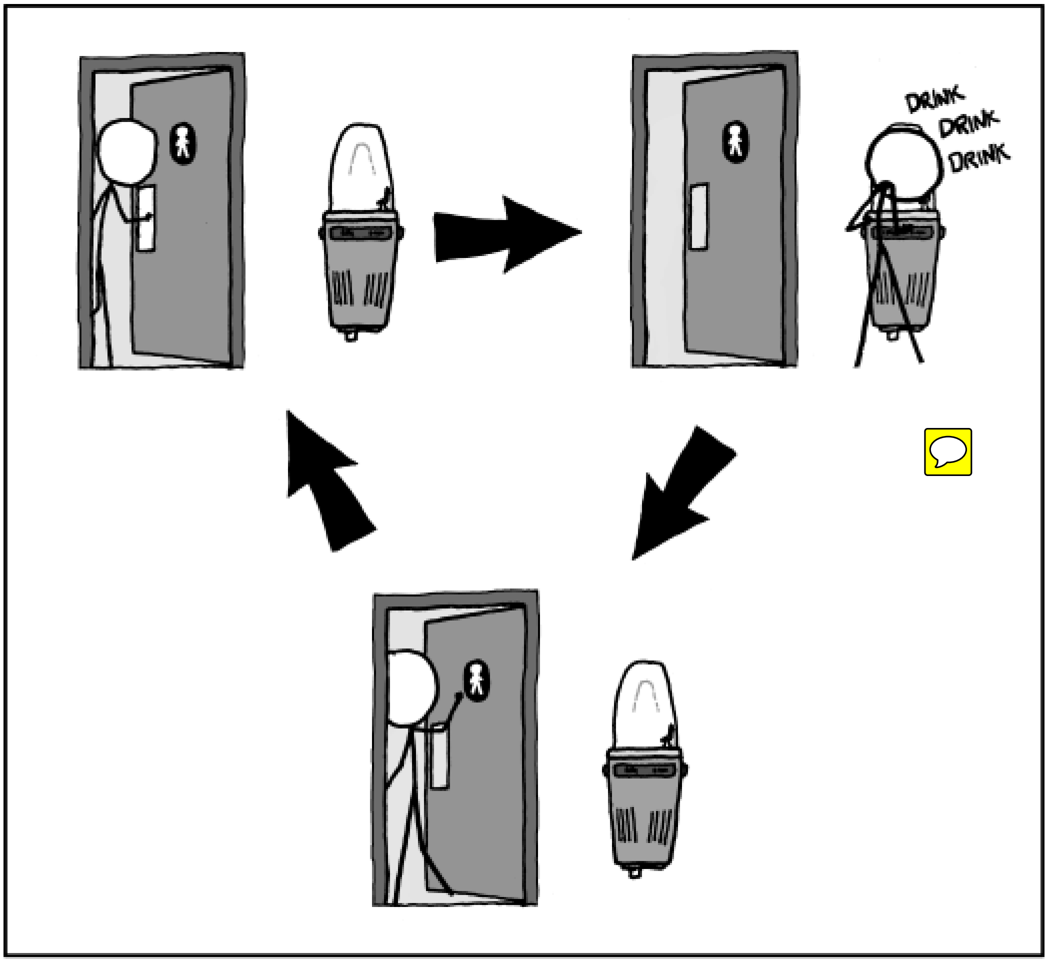
Writing a program without using GOTO instructions is an important rule in structured programming. A GOTO instruction points to a different part of the program without a guarantee of returning. Instead of using GOTO's, structures called "subroutines" or "functions" are used, which automatically return to the next instruction after the calling instruction when completed.


I COULD RESTRUCTURE THE PROGRAM'S FLOW
OR USE ONE LITTLE 'GOTO' INSTEAD.
EH, SCREW GOOD PRACTICE. HOW BAD CAN IT BE?
goto main_sub3;
COMPILE

Three fundamental control structures

The structured program theorem states that a given class of algorithms can be expressed using only three control structures:

Sequential	Decision (Selection)	Iterative
Refers to an ordered execution of statements	One of a number of statements is executed depending on the state of the program. This is usually expressed with keywords such as if..then..else..endif.	A statement is executed until the program reaches a certain state, or operations have been applied to every element of a collection. This is usually expressed with keywords such as while, repeat, for or do..until.



I AVOID DRINKING FOUNTAINS OUTSIDE BATHROOMS BECAUSE I'M AFRAID OF GETTING TRAPPED IN A LOOP.

Assignment – Control Structures

Using screenshots from scratch, present examples of the three control structures

Make 3 simple blocks of code in scratch that showcases each of the three control structures (Sequential, Decision, and Iterative).

Use the snipping tool to take the screenshots to then include in a separate document to hand in.

The first image clearly exemplifies the sequential control structures (2 marks)

The second image clearly exemplifies the decision (selection) control structure (2 marks)

The third image clearly exemplifies the iterative control structure (2 marks)

Save the document using the naming convention: LastName_FirstName_Assignment5