

Brief History of Programming

- ENIAC - Electronic Numerical Integrator and Calculator (1946)
 - first general-purpose computer

- Early computers used vacuum tubes for digital logic. These are large and emit tremendous amounts of heat. They are also sensitive.

- This didn't mesh well with programming a computer. Since this meant rewiring the vacuum tubes

- Each computer was different as to how the rewiring was done (lack of portability)

- Write Once - Run Anywhere

- This was the idea that a generic language for programming could be used. The program would only need to be written once.

- Each computer would provide a translator (compiler) that would translate the generic program into the specific needs of the computer

- Short Code - probably the first language in the modern sense. (1949)

- Flow-Matic - ultimately became COBOL in early 1960's (1955)
 - designed by Grace Hopper (U.S. Navy)

- Fortran - first to use modern compiler idea (1957)
 - designed by John Backus (IBM)

- There are now many languages (probably more than 200), but not all of these are used. Some are theoretical as opposed to practical.

Algorithms

16

- An algorithm is a set series of steps designed to achieve a goal
- Bad Algorithm 1

Lather
Rinse
Repeat

- The algorithm never ends if you interpret it literally (an infinite loop).

- computers interpret your instructions exactly as you specify.

- Infinite loops can occur in your own code. These can be hard to detect.

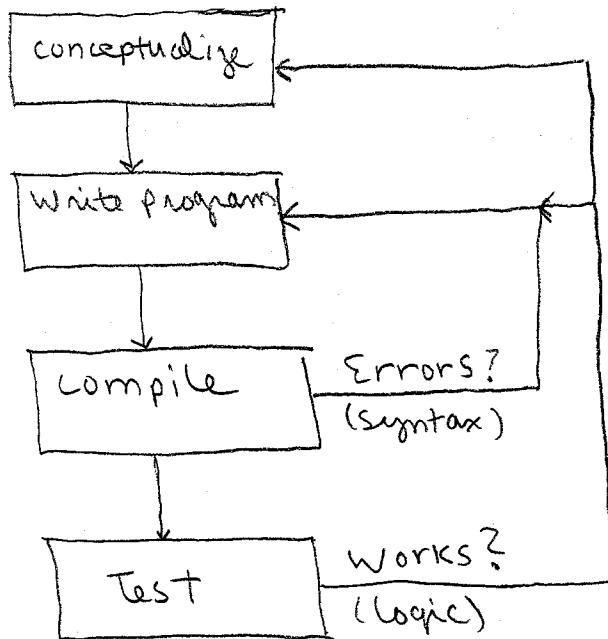
- (Kind of) Bad Algorithm 2

Brush Teeth
Drink orange Juice

- In this case, the ordering of the steps produces an undesirable result even though there is nothing technically "wrong"

- Algorithm for starting a 5 speed manual Transmission Car.

- Programs are basically algorithms
- Development of a program follows the scientific method



- Errors come in 2 main types
 - 1) Syntax errors. These occur when you have a statement in your program that the compiler can't translate
 - 2) Logic Errors. These occur when you give your program incorrect instructions.
- Syntax errors are easy to diagnose. Logic errors can be very tricky since the error may show up "far away" from where the error was made.
- The test phase is critical. Whenever feasible, a program should be run with a known set of inputs/outputs to see if it is producing correct results. This (usually) means the results are valid in cases where the results are not known ahead of time.