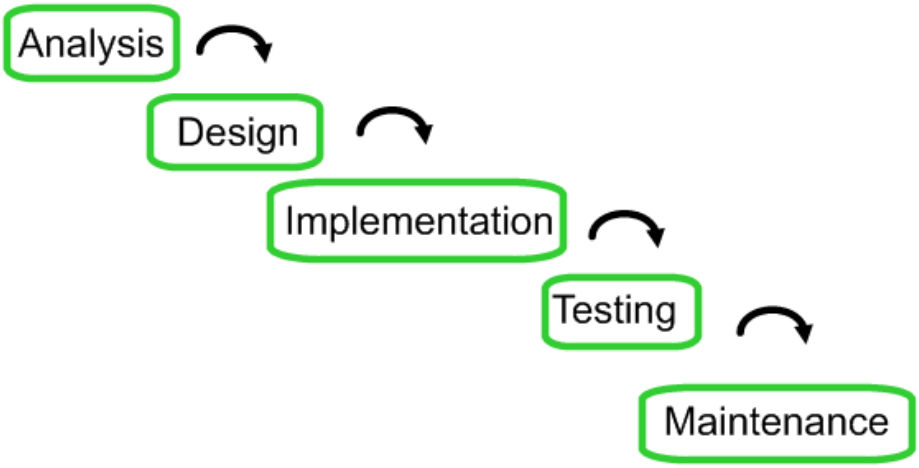
Program Life Cycle



Analysis Stage: Closely examining what the program must do
May include research, interviewing end users
Produces a *specification* (list of everything the program must do)
Written in plain English

Design Stage: Algorithms and design specifications are made
Written in plain English

Implementation Stage: Programmer takes each element of the design and translates it into source code

Testing Stage: -Trained Testers: Trained individuals attempt to crash the program
-Automated testing: A system set up that runs the program 1000's of times
-Beta Testing: Samples given out to end users to try out

Maintenance Stage: -Product ready for distribution
-Constant changes are made, updates, etc

Rule of Ten: A bug discovered in the implementation may cost on average a dollar to fix. Hen it is discovered in each subsequent stage it may cost ten times as much to fix.

Machine Language
Instructions to CPU's are given in Machine Language or Machine Code
Each type of CPU has it's own Machine Language
A Windows machine language program cannot be run on Linux or Apple system
Examples of machine language instructions:
Move a piece of data from RAM to register
Send a piece of data to a certain peripheral

Programming Language:
To avoid writing Machine Language, scientists developed programming language
Referred to as the source code

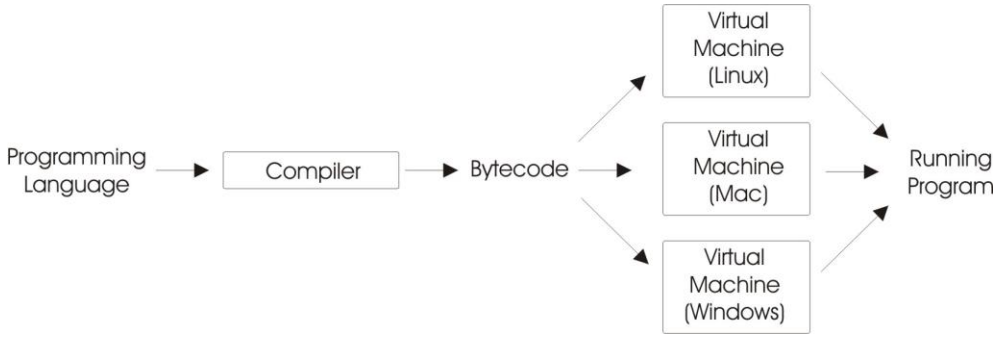
A *Compiler* translates the programming language into machine language
The translation is made ahead of time, before the program is run
The translated version of the program is then run

An *Interpreter* works differently from a compiler
An interpreter reads source code directly, and executes it a line at a time
It doesn't translate the source code ahead of time, it translates it at runt time, so running the program can take longer

Compiler-Interpreter
A system where both a compiler and an interpreter is used
First a program is compiled into **bytecode** (instead of machine language)
When the program is ready to be run, an interpreter interprets each line of byte code, one at a time.

In a compiler-interpreter the byte code is machine-independent; it can be interpreted using any CPU and any system (Windows, Linux, Apple, etc)

JAVA is the best known compiler-interpreter system.
Its interpreter is called JAVA Virtual Machine



JAVA

- Originally named OAK.
- Created in the 1990's to control set-top TV boxes.
- It used a compiler-interpreter model to provide portability

PROGRAMMING ERRORS

Compile Time Errors:

An error in the format of the instruction.
As a result the program cannot be compiled.
Ex: Move the airplane blue
There is a syntax error

Run-Time Errors:

Program executes but terminates unexpectedly (crash)
Ex: Travel 10 km then turn right at McDonald's
If there was no McDonald's then you cannot perform the instruction
Reading from a file that doesn't exist, divide a number by zero

Logic Errors:

Program executes but produces incorrect results