Deployment and Operations for Software Engineers 2nd Ed

Chapter 8—DevOps preliminaries

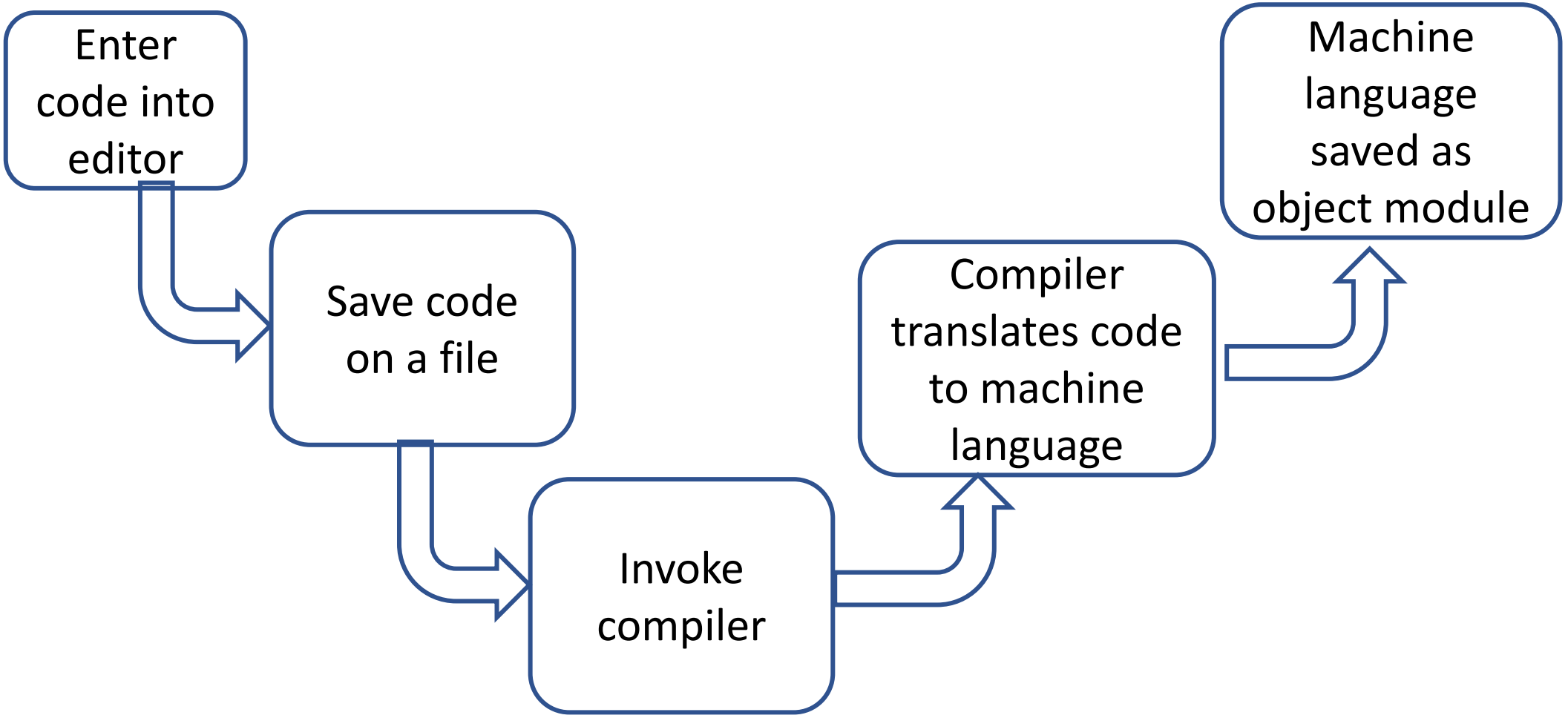
Outline

- Constructing an executable
- Invoking an executable
- Imperative vs declarative languages
- Modifiability

Executable image

- An executable image is a set of machine language instructions
  - Represented as a set of bits on a disk
  - Can be loaded by an OS.
  - Control can be transferred to it.
- Consists of libraries and code you wrote
- Constructed differently for compiled and interpreted languages.

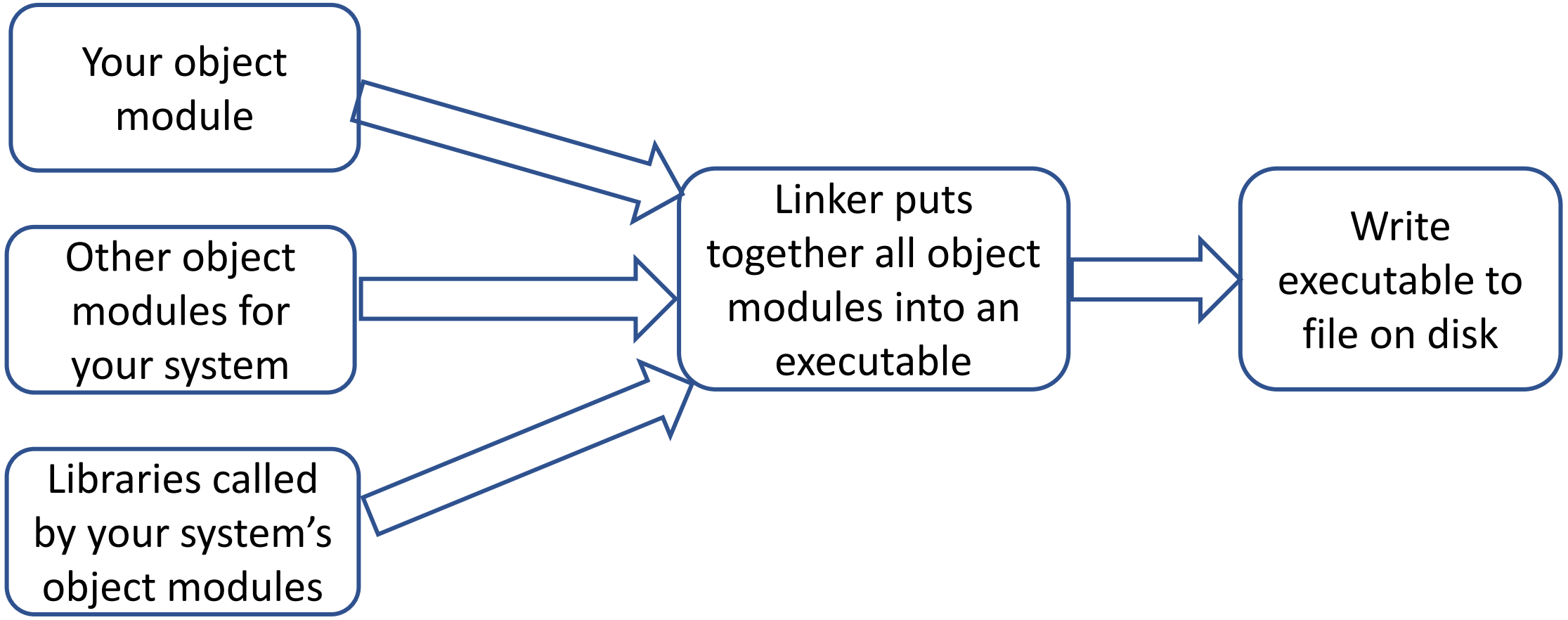
Compiled language => Object module



©Len Bass and John Klein 2022

4

Object Modules => Executable



Interpreted language => Executable

Your code in interpreted language
written to file

Executable version of interpreter runtime including libraries on disk file

Invoking executable image

- Prequisites
  - Your executable on $file_1$
  - Possibly your interpreter code on $file_2$
  - Data for your system on $file_3$
- Steps
  - Human or script issues command to Command Line Interpreter (CLI): “Execute $file_1$ [$file_2$]with $file_3$”
  - CLI calls OS. Execute $file_1$ [$file_2$]with $file_3$”
  - OS loads $file_1$, transfers control to it with parameter of [$file_2$], $file_3$

Imperative language

- Specify steps computer goes through
- Steps are a series of state changes
- Must be debugged to ensure that series of state changes ends up with desired result
- Examples: C, C++, Java

Declarative language

- Specify desired end state of computer
- Language interpreter will determine how to achieve desired result.
- Must be examined to determined if achieved result is one desired.
- Examples: HTML, SQL

Modifiability

- A system is modifiable if changes to that system are easy to make.
- You want to make changes without side effects. A side effect is an impact that you did not intend.
- Modifiable systems are achieved with low coupling and high cohesion

Coupling and cohesion

- Coupling is a measure of how much overlap there is between two independent modules. Overlap can lead to unanticipated side effects.
- Cohesion is a measure of how strongly the responsibilities of a module are related. If all responsibilities have the same goal, cohesion is high. If some responsibilities are oriented toward one goal and others toward another, cohestion is low and side effects are possible.