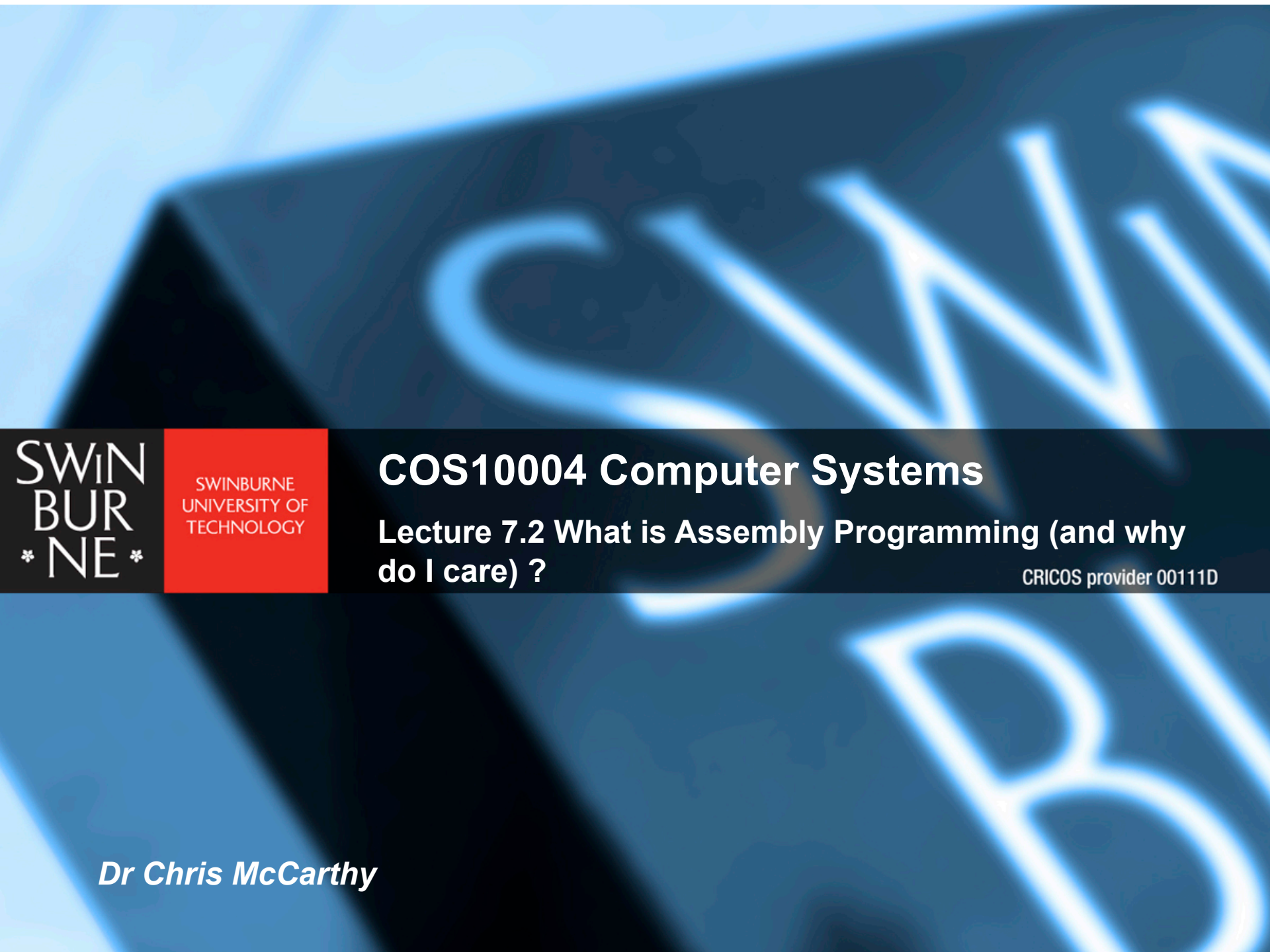


COS10004 Computer Systems

Lecture 7.2 What is Assembly Programming (and why do I care) ?

CRICOS provider 00111D

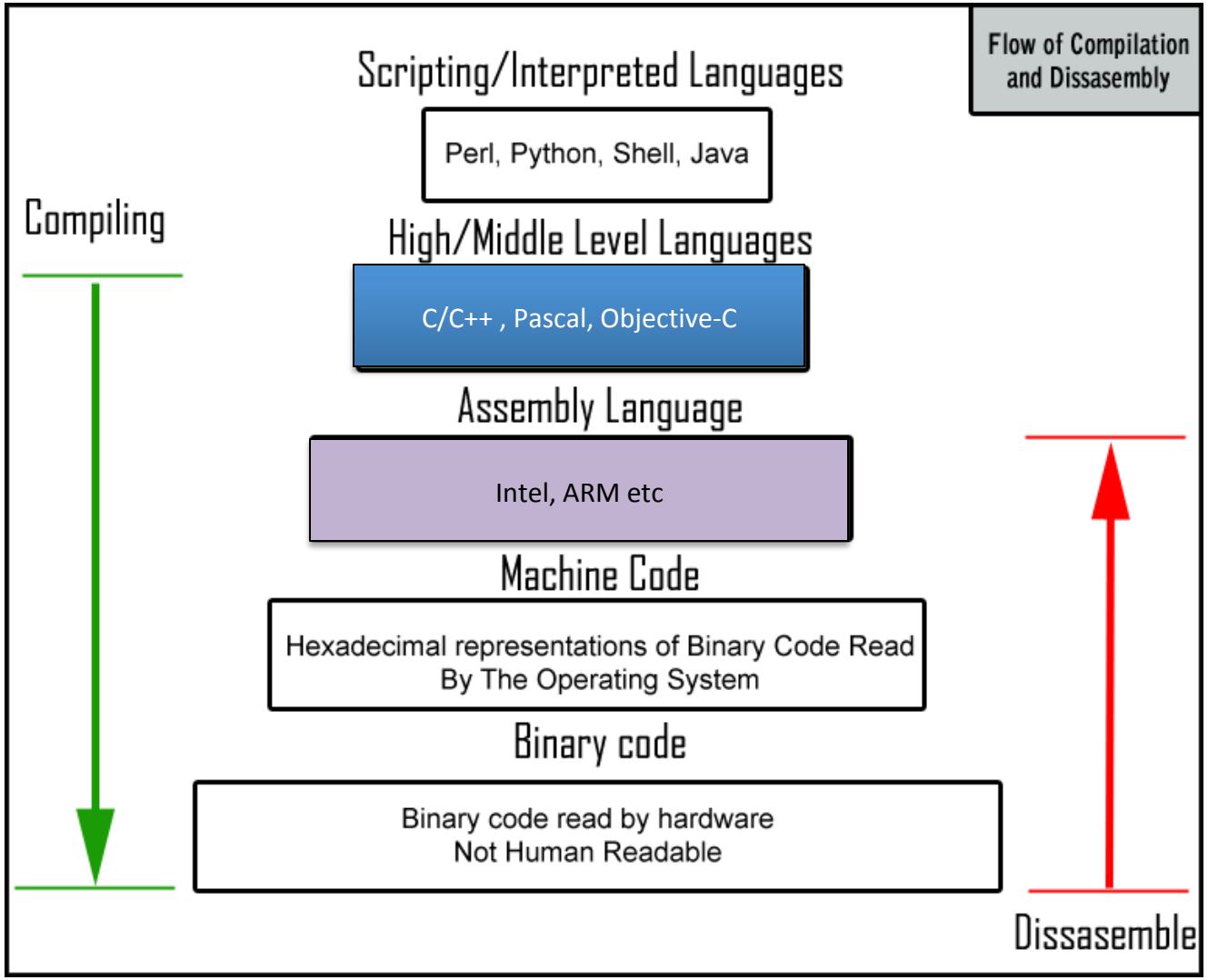
Dr Chris McCarthy

_start:

WHAT IS ASSEMBLY PROGRAMMING ?

- All computer processors:
  - Manages arithmetic, logic and control
- Different processor families have specific languages :
  - Eg Intel Core, Advanced RISC Machine (ARM) etc
- Lowest level of human readable programming
- The “assembler” translates code into binary machine instructions
  - Essentially bit strings that are interpretable by the processor

PROGRAMMING LANGUAGE HIERARCHY



Adapted from: https://blog.malwarebytes.com/security-world/2012/09/so-you-want-to-be-a-malware-analyst/

Assembly – Why ?

- Why do you need to know it?
  - Write tiny / really fast programs
  - No operating system to get in the way
  - Write compilers
  - Write drivers for custom hardware
  - Find vulnerabilities in code.

ASSEMBLY – WHY ?

- Why do you need to know it?
  - Reverse-engineering binaries
  - Eg., C/C++ compile to native machine code.
  - No comments, variable names, control structures (if, loops). Really hard to figure out what is going on.
- All binaries can be disassembled to ASM!

RISC VERSUS CISC

- Processor instruction sets can be classified as CISC or RISC
- RISC: Reduced Instruction Set Computer
  - Eg. ARM
- CISC: Complex Instruction Set Computer
  - Eg. Intel

RISC VERSUS CISC

CISC – Complex Instruction Set	RISC – Reduced Instruction Set
Emphasis on hardware	Emphasis on software
Includes multi-clock complex (i.e. specialised instructions)	Single-clock, reduced instruction only
Small code size	Large code sizes
Instructions have variable cycle time	Typical take one cycle
More transistors typically used for storing complex instructions	Less transistors, typically used more for memory registers
Higher power usage	Lower power usage
Typically well suited to multimedia applications	Typically well suited to low powered contexts (eg., mobile phones)

ARM ASSEMBLY PROGRAMMING

- Arm: Advanced RISC Machine
- The most widely used instruction set in the world:
  - 130 billion ARM processors produced
  - mobile devices, routers, IoT devices (eg RPi, Arduino)
- Consists of about 100 instructions in total:
  - We will explore some but certainly not all

SUMMARY

- ASM programming represents the lowest level of human readable programming
- Near 1-to-1 match with machine instructions
- RISC versus CISC
- ASM programming is useful for:
  - Hardware specific programming
  - Optimising code for performance
  - Reverse engineering and analysing code behaviour