Module 0: Introduction

Ron K. Cytron

Department of Computer Science and Engineering
Washington University in Saint Louis
Thanks to Alan Waldman
for comments that improved these slides

Prepared for 2u
Semester Online

Copyright Ron K. Cytron 2013
0.0 Introduction

• Why study computer science?
  – Isn’t this just programming?
  – Isn’t this just for computer scientists?

• Why take this course?
  – Logical thought processes
  – Computational literacy

• What do I need?
  – Computer
    • Mac or PC
    • Mac OS X, Windows, Linux
  – Repository
    • Set up using instructions

• What should I know already?
  – Basic understanding of math (algebra)
  – No prior programming experience necessary
0.1 A Generation of Computation

- Grandparent gives 7-year-old grandchild a gift
  - I got DigiComp
0.1 A Generation of Computation

• Grandparent gives 7-year-old grandchild a gift
  – I got DigiComp
  – My daughter got Furby
0.1 A Generation of Computation

• Grandparent gives 7-year-old grandchild a gift
  – I got DigiComp
  – My daughter got Furby

• Let’s compare these two toys
0.1 A Generation of Computation

- Cost
  - $60
- Input
  - 3 sliders
  - Each 0 or 1
- Output
  - 3 sliders, each 0 or 1

- Cost
  - $35
- Input
  - Eyes, ears, touch
  - Middle ear
- Output
  - Motion, speech
0.1 A Generation of Computation

- **Storage**
  - 3 bits (8 integers)

- **Speed**
  - 1 cycle / second
  - 0.000001 MIPS

- **Reliability**
  - Breaks every 10 seconds

- **Storage**
  - 128 Megabits (16 songs)

- **Speed**
  - 1 cycle / 62 nanosecs
  - 16 MIPS

- **Reliability**
  - Breaks every 5 years
0.2 Thirty Years’ Improvement

• Consider a personal computer you might have bought in 1980
0.2 Thirty Years’ Improvement

• And now consider a personal computer you might have bought 30 years later
0.2 Thirty Years’ Improvement

- **Cost (today’s $$)**
  - $13,000
- **Speed**
  - 0.4 MIPS
- **Input**
  - Keyboard, serial port

- **Cost**
  - $1,000
- **Speed**
  - > 2,000 MIPS
- **Input**
  - Keyboard, mouse, microphone
0.2 Thirty Years’ Improvement

- **Output**
  - Character-only display
  - Beeps
- **Storage (brain)**
  - 64 Kbytes
- **Reliability**
  - Breaks every 6 months

- **Output**
  - Bitmapped screen
  - HD Sound
- **Storage (brain)**
  - 4,000,000 Kbytes
- **Reliability**
  - Breaks every 6 years
0.2 Thirty Years’ Improvement

- **Long-term storage**
  - 5 Mbytes
- **Weight**
  - 55 pounds
- **Lifetime unplugged**
  - 0 hours

- **Long-term storage**
  - 256,000 Mbytes
- **Weight**
  - 3 pounds
- **Lifetime unplugged**
  - 12 hours
0.2 Thirty Years’ Improvement

What if the same improvement could happen to a car in 30 years?
0.2 Thirty Years’ Improvement

- Cost
  - $4,000
- Speed
  - 60,000 mph
- Seating
  - 10,000 people
0.2 Thirty Years’ Improvement

- Fuel efficiency
  - 20,000 mpg
- Reliability
  - Breaks every 70 years
0.3 What is computer science?

• A discipline with deep intellectual roots
  – Math, logic, philosophy, engineering

• Computation
  – “Computors” were people who computed things
    • Tables of function values: sine, cosine, log, etc.
  – 1823: Babbage’s Difference Engine
  – 1937: Turing proves the limits of computation
    • Turing Machine remains best model of computation to this day

• Computers help us solve important problems
  – Health
    • Genetic mapping, sequencing, phylogeny
    • Drug design and interactions
  – Environment
    • Climate and weather prediction, simulation instead of construction
  – Energy
    • Finding resources, conservation
0.3 What is computer science?

• A discipline with broad impact
  – Art, music
  – Science, engineering
  – Business, commerce

• A people-facing discipline
  – Software development is a collaborative activity
  – Research at the interface of human and computer
  – Literate programming

• A discipline that changes how you think
  – Logical approach to problem solving
  – Designs that facilitate automation
0.4 How to succeed in this course

• Keep up with readings and lecture material
• For studios and labs, find collaborators whose place feels comfortable to you
  – Ask questions of the course staff
    • Professor
    • Teaching assistants
• For lab and extension assignments, try to solve these on your own
  – But ask for help when you are stuck
  – Wrestling with difficulties is more instructive than having the correct solution immediately