Welcome to Bristol!
COMSM1302 Overview of Computer Architecture

John Lapinskas, University of Bristol

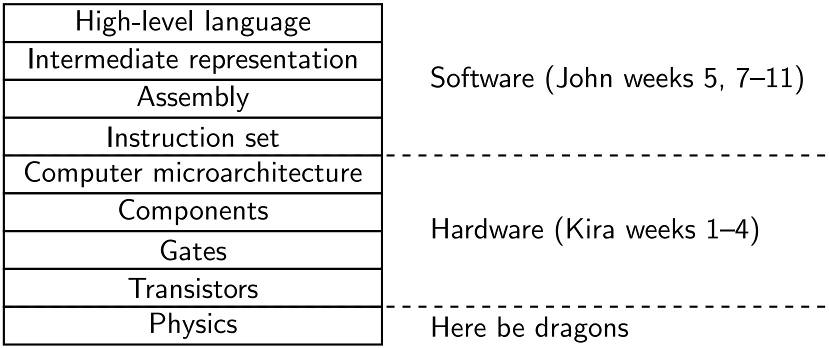
What is architecture?

Programming in C starts out with a “Hello, world!” program...

What is architecture?

Programming in C starts out with a “Hello, world!” program...

but there’s a lot going on under the surface!


High-level language
Intermediate representation
Assembly
Instruction set
Computer microarchitecture
Components
Gates
Transistors
Physics
Software (John weeks 5, 7–11)
Hardware (Kira weeks 1–4)
Here be dragons

This unit is about filling in that gap and understanding what’s going on between writing a program in C and having it execute on your PC.

Why bother?

Normally, we survive this complexity with **abstraction** — focusing only on the level we absolutely need. So why bother learning architecture if you'll mostly be programming high-level languages?

Why bother?

Normally, we survive this complexity with **abstraction** — focusing only on the level we absolutely need. So why bother learning architecture if you'll mostly be programming high-level languages?

- Working with low-power/embedded systems.

Why bother?

Normally, we survive this complexity with **abstraction** — focusing only on the level we absolutely need. So why bother learning architecture if you'll mostly be programming high-level languages?

- Working with low-power/embedded systems.
- Writing highly-optimised code.

Why bother?

Normally, we survive this complexity with **abstraction** — focusing only on the level we absolutely need. So why bother learning architecture if you'll mostly be programming high-level languages?

- Working with low-power/embedded systems.
- Writing highly-optimised code.
- Understanding and avoiding coding traps and pitfalls.

Why bother?

Normally, we survive this complexity with **abstraction** — focusing only on the level we absolutely need. So why bother learning architecture if you'll mostly be programming high-level languages?

- Working with low-power/embedded systems.
- Writing highly-optimised code.
- Understanding and avoiding coding traps and pitfalls.
- Don't you want to know how to build a computer from scratch?

Why bother?

Normally, we survive this complexity with **abstraction** — focusing only on the level we absolutely need. So why bother learning architecture if you'll mostly be programming high-level languages?

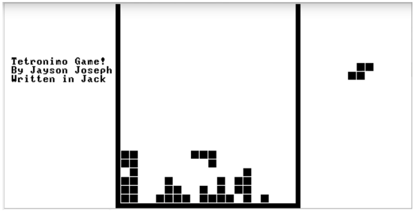
- Working with low-power/embedded systems.
- Writing highly-optimised code.
- Understanding and avoiding coding traps and pitfalls.
- Don't you want to know how to build a computer from scratch?

By the end of this unit, you'll have built a working CPU and created a compiler for a high-level language.

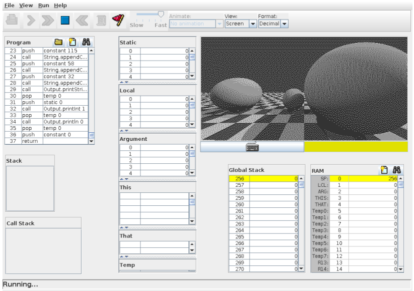
It won't be as powerful as C, but it will be powerful enough to demystify C and get a good sense of what the C compiler is actually doing.

Hack and Nand2Tetris

The CPU we'll be building runs on an architecture called Hack, designed to be simple enough for educational use but powerful enough to run Tetris.



Source: Jayson Joseph (here)



```
23  push    constant 115
24  call    String.appendC...
25  push    constant 58
26  call    String.appendC...
27  push    constant 32
28  call    String.appendC...
29  call    Output.printStri...
30  pop     temp 0
31  push    static 0
32  call    Output.printInt 1
33  pop     temp 0
34  call    Output.println 0
35  pop     temp 0
36  push    constant 0
37  return
```

Global Stack	
256	0
257	0
258	0
259	0
260	0
261	0
262	0
263	0
264	0
265	0
266	0
267	0
268	0
269	0
270	0

RAM		
SP:	0	256
LCL:	1	0
ARG:	2	0
THIS:	3	0
THAT:	4	0
Temp0:	5	0
Temp1:	6	0
Temp2:	7	0
Temp3:	8	0
Temp4:	9	0
Temp5:	10	0
Temp6:	11	0
Temp7:	12	0
R13:	13	0
R14:	14	0

Running...

Source: Alex Quach (here)

We very loosely follow the Nand2Tetris model, originally developed for Herzliya and Jerusalem universities and subsequently released free online.

Both courses build up a Hack CPU in a similar fashion, but apart from that they're quite different — we cover things they don't and vice versa.

The bad news: What we expect from you

This is a hard degree programme with a significant failure rate.

You should put in $\sim$40 hours/week total, and $\sim$15 hours/week in this unit.

The unit runs as a flipped classroom. Every Friday evening, we release video lectures and an assignment for the following week, which will take most of your time. The following Monday, we meet for a problem-solving session on the **previous** week's material.

You should aim to arrive on Monday already knowing the $\underline{\text{previous}}$ week's material and having stopped work on the $\underline{\text{previous}}$ week's assignment.

The bad news: What we expect from you

This is a hard degree programme with a significant failure rate.

You should put in $\sim$40 hours/week total, and $\sim$15 hours/week in this unit.

The unit runs as a flipped classroom. Every Friday evening, we release video lectures and an assignment for the following week, which will take most of your time. The following Monday, we meet for a problem-solving session on the **previous** week's material.

You should aim to arrive on Monday already knowing the previous week's material and having stopped work on the previous week's assignment.

You might not be able to finish the assignments, and that's OK — we've erred towards making them longer so you have more revision material.

But you should always make sure you're putting your 15 hours in. Each week builds on the last, so while we do release solutions, if you fall behind then you're in trouble.

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

- Four hours of TA-supported labs split Wednesday/Thursday.
    - Drop-in/drop-out as needed.
    - Great for working on assignments!

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

- Four hours of TA-supported labs split Wednesday/Thursday.
    - Drop-in/drop-out as needed.
    - Great for working on assignments!
- One hour Q&A on Tuesdays.
    - Padlet-supported so you can ask anonymously!

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

- Four hours of TA-supported labs split Wednesday/Thursday.
    - Drop-in/drop-out as needed.
    - Great for working on assignments!
- One hour Q&A on Tuesdays.
    - Padlet-supported so you can ask anonymously!
- One hour office hours on Wednesdays.

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

- Four hours of TA-supported labs split Wednesday/Thursday.
    - Drop-in/drop-out as needed.
    - Great for working on assignments!
- One hour Q&A on Tuesdays.
    - Padlet-supported so you can ask anonymously!
- One hour office hours on Wednesdays.
- Hour-long drop-in sessions with TAs (rooms and times TBD).

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

- Four hours of TA-supported labs split Wednesday/Thursday.
    - Drop-in/drop-out as needed.
    - Great for working on assignments!
- One hour Q&A on Tuesdays.
    - Padlet-supported so you can ask anonymously!
- One hour office hours on Wednesdays.
- Hour-long drop-in sessions with TAs (rooms and times TBD).
- Teams channel to ask questions of TAs/lecturers at any time.

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

- Four hours of TA-supported labs split Wednesday/Thursday.
  - Drop-in/drop-out as needed.
  - Great for working on assignments!
- One hour Q&A on Tuesdays.
  - Padlet-supported so you can ask anonymously!
- One hour office hours on Wednesdays.
- Hour-long drop-in sessions with TAs (rooms and times TBD).
- Teams channel to ask questions of TAs/lecturers at any time.
- Recommended readings from textbooks (available for free from UoB).
  - You're **not** expected to read these, but sometimes you just need to see something explained in a different way.

The good news: What you can expect from us

The good news is: we also give you a lot of support! This is all technically optional, but we really strongly recommend taking advantage of it.

- Four hours of TA-supported labs split Wednesday/Thursday.
  - Drop-in/drop-out as needed.
  - Great for working on assignments!
- One hour Q&A on Tuesdays.
  - Padlet-supported so you can ask anonymously!
- One hour office hours on Wednesdays.
- Hour-long drop-in sessions with TAs (rooms and times TBD).
- Teams channel to ask questions of TAs/lecturers at any time.
- Recommended readings from textbooks (available for free from UoB).
  - You're **not** expected to read these, but sometimes you just need to see something explained in a different way.
- Online resources — Hack is popular with hobbyists!
  - Avoid ChatGPT though, it's often confidently wrong.

Assessment and exams

- The unit will have two in-person two-hour exams.
  - The first exam will take place Friday of week 6 and be worth 40%.
  - The second exam will take place Thursday of week 13 and be worth 60%.
- The questions will go from very easy to very hard. You probably won't finish, but the time limit is very generous if you're after a pass or merit and doable for a distinction.

Assessment and exams

- The unit will have two in-person two-hour exams.
  - The first exam will take place Friday of week 6 and be worth 40%.
  - The second exam will take place Thursday of week 13 and be worth 60%.
- The questions will go from very easy to very hard. You probably won't finish, but the time limit is very generous if you're after a pass or merit and doable for a distinction.
- Both exams will be half-theory and half-practical.
  - The theory part will be an auto-marked Blackboard quiz and will be similar to the Monday live sessions.
  - The practical part will be based around your assignments — designing circuits for the first exam, writing assembly code for the second.

Assessment and exams

- The unit will have two in-person two-hour exams.
    - The first exam will take place Friday of week 6 and be worth 40%.
    - The second exam will take place Thursday of week 13 and be worth 60%.
- The questions will go from very easy to very hard. You probably won't finish, but the time limit is very generous if you're after a pass or merit and doable for a distinction.
- Both exams will be half-theory and half-practical.
    - The theory part will be an auto-marked Blackboard quiz and will be similar to the Monday live sessions.
    - The practical part will be based around your assignments — designing circuits for the first exam, writing assembly code for the second.
- You'll have access to a reference sheet, but not unrestricted Internet access or lecture notes for the whole unit.

Assessment and exams

- The unit will have two in-person two-hour exams.
  - The first exam will take place Friday of week 6 and be worth 40%.
  - The second exam will take place Thursday of week 13 and be worth 60%.
- The questions will go from very easy to very hard. You probably won't finish, but the time limit is very generous if you're after a pass or merit and doable for a distinction.
- Both exams will be half-theory and half-practical.
  - The theory part will be an auto-marked Blackboard quiz and will be similar to the Monday live sessions.
  - The practical part will be based around your assignments — designing circuits for the first exam, writing assembly code for the second.
- You'll have access to a reference sheet, but not unrestricted Internet access or lecture notes for the whole unit.

If you have AEAs (e.g. extra time), please get in touch with us early! We don't make decisions on who gets AEAs (the Disability Office does), but it'll be really helpful to have an idea of numbers for week 6.

Bristol culture

- Lecturers are generally easy-going and hard to offend by accident.
- We **like** getting questions!
  - But please put them on the unit Team — that way everyone gets to benefit from the answer. There's no shame in needing help, we all remember finding this material difficult ourselves.
  - If you'd still rather ask anonymously, that's what the office hours, Q&As and drop-ins are for — emails to us go to the back of the queue.

Bristol culture

- Lecturers are generally easy-going and hard to offend by accident.
- We **like** getting questions!
    - But please put them on the unit Team — that way everyone gets to benefit from the answer. There's no shame in needing help, we all remember finding this material difficult ourselves.
    - If you'd still rather ask anonymously, that's what the office hours, Q&As and drop-ins are for — emails to us go to the back of the queue.
- If you see a typo, please don't hesitate to ask publicly!
    - If it's real we'll need to tell people anyway, and if it's not then you won't be the only one confused.

Bristol culture

- Lecturers are generally easy-going and hard to offend by accident.
- We **like** getting questions!
    - But please put them on the unit Team — that way everyone gets to benefit from the answer. There's no shame in needing help, we all remember finding this material difficult ourselves.
    - If you'd still rather ask anonymously, that's what the office hours, Q&As and drop-ins are for — emails to us go to the back of the queue.
- If you see a typo, please don't hesitate to ask publicly!
    - If it's real we'll need to tell people anyway, and if it's not then you won't be the only one confused.
- We take academic misconduct and plagiarism very seriously.

Bristol culture

- Lecturers are generally easy-going and hard to offend by accident.
- We **like** getting questions!
  - But please put them on the unit Team — that way everyone gets to benefit from the answer. There's no shame in needing help, we all remember finding this material difficult ourselves.
  - If you'd still rather ask anonymously, that's what the office hours, Q&As and drop-ins are for — emails to us go to the back of the queue.
- If you see a typo, please don't hesitate to ask publicly!
  - If it's real we'll need to tell people anyway, and if it's not then you won't be the only one confused.
- We take academic misconduct and plagiarism very seriously.
- First names are always fine. Honorifics (e.g. "Dr. Lapinskas") are also always fine and a bit more formal. *Wrong* honorifics (e.g. "Mr. Lapinskas") will usually make someone roll their eyes a bit.

A warning

Early in Programming in C, you will run into “pointers”.

They will look disconnected from what you’ve learned so far, especially if you have previous programming experience in a higher-level language like Python. They will also be hard to understand.

You will be tempted to ignore them and wait until the unit moves on.

A warning

Early in Programming in C, you will run into “pointers”.

They will look disconnected from what you’ve learned so far, especially if you have previous programming experience in a higher-level language like Python. They will also be hard to understand.

You will be tempted to ignore them and wait until the unit moves on.

DO NOT DO THIS!

Pointers are the standard example of memory indirection, which is vital not just to C, but to every programming language under the hood.

C is not going to move on from pointers. C is going to keep using them for the whole term. Then Architecture is going to use them (or the idea behind them) for the whole second half as well!

Good luck!

Any questions?