

Coding Your Future: A Guidebook for Students

Duncan Hull and illustrated by Bryan Mathers

Contents

Welcome to your future	9
0.1 Imagining your future	10
0.2 Your future aims	10
0.3 What you won't learn	12
0.4 Learning your future	12
0.5 Mapping your future	14
0.6 Your future themes	17
0.7 Downloading your future	18
0.8 Contributing to your future	18
0.9 Acknowledgements	21
0.10 About me	56
0.11 Legal stuff	57
I DESIGNING	61
1 Rebooting your future	63
1.1 What you will learn	63
1.2 Let's go down the rabbit hole	65
1.3 Opening your future	65
1.4 Maximising your future	66
1.5 Your future is your responsibility	66
1.6 Your degree is not enough	66
1.7 It's too late when you graduate	70
1.8 Yes, this WILL be on the exam	72
1.9 Practicing your future	74
1.10 Navigating your future	74
1.11 Crediting your future	74
1.12 Your future is different	75
1.13 Engaging with your future	81
1.14 Signposting your future	82
1.15 Summarising your future	82

2 Knowing your future	85
2.1 What you will learn	85
2.2 Improving self-awareness	85
2.3 Breakpoints	98
2.4 Signposts from here on identity	98
2.5 Summarising self awareness	101
3 Nurturing your future	105
3.1 What you will learn	105
3.2 Stressed, depressed and exam obsessed?	105
3.3 Mental ill health	107
3.4 Look after yourself	115
3.5 Burnout	116
3.6 Help is available if you need it	118
3.7 Developing a growth mindset	124
3.8 Wellbeing signposts	126
3.9 Breakpoints	128
3.10 Summarising well-being	129
4 Writing your future	131
4.1 What you will learn	131
4.2 Writing as a soft social skill	133
4.3 Computing is your superpower!	134
4.4 Communication I/O	135
4.5 Writing your future	142
4.6 Breakpoints	146
4.7 Summarising your soft skills	147
5 Computing your future	151
5.1 What you will learn	151
5.2 Computing is for everybody	151
5.3 Software is eating your future	154
5.4 Computing is eating the world	155
5.5 Passive consumer or active producer?	157
5.6 Play your joker: Computational joker	157
5.7 Summarising computing your future	157
II TESTING & BUILDING	163
6 Debugging your future	165
6.1 What you will learn	165
6.2 Beware of the black hole	167
6.3 The CV is dead, long live the CV!	168
6.4 It's not a bug, it's a feature!	169
6.5 Is it a bug or a feature?	171

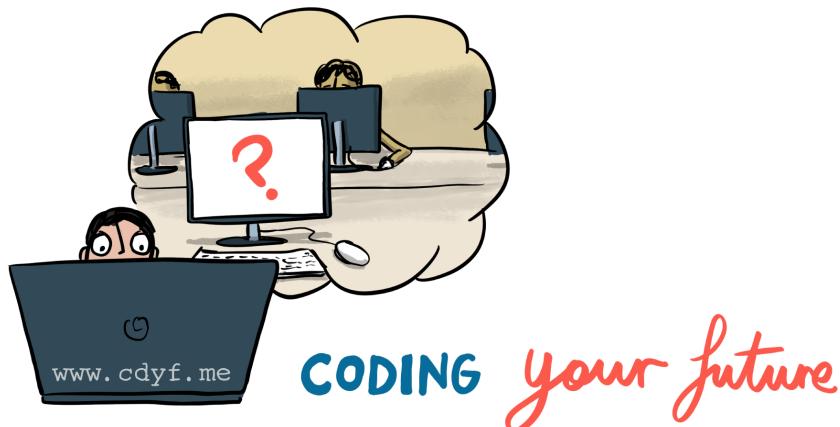
6.6 Structure your CV	171
6.7 Birds eye view	184
6.8 Breakpoints	194
6.9 Checklist: Big Bad Bugs	195
6.10 Covering letters & personal statements	196
6.11 Debugging summary	198
7 Experiencing your future	201
7.1 What you will learn	201
7.2 Why is experience so valuable?	201
7.3 Are you experienced?	204
7.4 Breakpoints	213
7.5 Summarising your experience	214
8 Finding your future	215
8.1 What you will learn	215
8.2 Job search strategies	217
8.3 Where can you look for jobs?	226
8.4 Buyer beware	233
8.5 Breakpoints	242
8.6 Summarising search	243
9 Broadening your future	245
9.1 What you will learn	247
9.2 Beyond software engineering	247
9.3 Research software engineering	248
9.4 With great code comes great responsibility	250
9.5 Do you need to sell your soul?	251
9.6 Computing the future	253
9.7 Breakpoints	254
9.8 Summarising your alternatives	256
10 Speaking your future	259
10.1 What you will learn	261
10.2 Interviews	261
10.3 Breakpoints	268
10.4 Accepting job offers	270
10.5 Communication, communication, communication!	270
10.6 Summarising interviews	271
11 Surviving your future	273
11.1 What you will learn	273
11.2 Survive, thrive or dive?	273
11.3 Managing your manager	277
11.4 Stay in school	277
11.5 Breakpoints	279

11.6 Reflecting your future	280
11.7 Summarising survival	280
12 Achieving your future	283
12.1 What you will learn	283
12.2 Academic badges	283
12.3 Digital badges	286
12.4 Other digital badges	291
12.5 Breakpoints	291
12.6 Summarising your achievements	292
13 Researching your future	295
13.1 What you will learn	296
13.2 Where to start	296
13.3 University or the Real World?	296
13.4 Ask beyond academia	297
13.5 Breakpoints	297
13.6 Signposts from here on research	298
13.7 Summarising further study and research	298
III DEPLOYING	301
14 Ruling your future	303
14.1 Know who you are	303
14.2 Look after yourself	305
14.3 Use what you have	305
14.4 Grow your networks	307
14.5 Always make new mistakes	307
14.6 Help and thank who you can	310
14.7 Look beyond the obvious	310
14.8 Stay in school	311
14.9 Step outside your comfort zone	311
14.10 Don't give up	313
14.11 Ten simple summaries	313
15 Hacking your future	315
15.1 Hack their CVs	315
15.2 Breakpoints	316
15.3 Sample CVs	317
15.4 Sample CoolTech Job advert	320
16 Moving your future	321
16.1 Hit the South?	321
16.2 Hit the North!	322
16.3 Hit the East and West	322

CONTENTS	7
16.4 Summarising moving your future	322
17 Enjoying your future	325
17.1 The Coder’s playlist	326
17.2 Turing’s electro-playlist	327
17.3 The Lancashire playlist	329
17.4 The Dad-rock playlist	329
17.5 The Indie rock playlist	332
17.6 Thanks Jim	333
18 Actioning your future	335
18.1 Your actions define your impact	335
18.2 What you will learn	335
18.3 Breakpoints	336
18.4 Team verbs	337
18.5 Engineering verbs	337
18.6 Leadership verbs	338
18.7 Improving verbs	338
18.8 Scientific verbs	339
18.9 Winning verbs	339
18.10 Organising verbs	340
18.11 Influential verbs	340
18.12 Summarising your actions	340
19 Scheduling your future	343
19.1 Week 11: Meeting	343
19.2 Week 12: Meeting	343
19.3 Discussing your Future	345
IV CODING	347
20 Hearing your future	349
20.1 Subscribing to your future	349
20.2 Episode 5: Sneha Kandane	351
20.3 Episode 4: Carmen Faura Práxedes	351
20.4 Episode 3: Brian Yim Tam	353
20.5 Episode 2: Jason Ozuzu	353
20.6 Episode 1: Raluca Cruceru	353
20.7 Episode <i>x</i> : it could be YOU!	354
20.8 Everything I had to know, I heard it on my radio	360
21 Raluca	363
21.1 What’s your story Raluca?	363
21.2 Web Applications UK	365
21.3 CERN	366

21.4 Final year	370
21.5 Advice to your former self	372
21.6 The future	374
21.7 Disclaimer	375
22 Brian	377
22.1 What's your story Brian?	377
22.2 Disclaimer	377
23 Jason	379
23.1 What's your story Jason?	379
23.2 Morgan Stanley	382
23.3 Google	384
23.4 Final year	386
23.5 Minority report	388
23.6 One tune	392
23.7 One podcast	393
23.8 One film	393
23.9 Advice to your former self	394
23.10 Disclaimer	395
24 Carmen	397
24.1 What's your story Carmen?	397
24.2 The Walt Disney Company	400
24.3 McKinsey	401
24.4 One tune	407
24.5 One podcast	407
24.6 One film	407
24.7 Advice to your former self	408
24.8 Disclaimer	409
25 Sneha	411
25.1 What's your story Sneha?	411
25.2 Matillion	414
25.3 One tune	420
25.4 One podcast	421
25.5 One film	421
25.6 Advice to your former self	422
25.7 Disclaimer	423
26 Alice	425
26.1 What's your story Alice?	425
27 Reading your future	427

Welcome to your future



Hello and welcome to *Coding Your Future* at www.cdyf.me. This guidebook will help you to design, build, test and **code** your future in computing. Also available as a free ebook and pdf (see section 0.7), this guide is aimed at ALL students in higher education. While the guide supports undergraduate teaching at the University of Manchester, it doesn't actually matter:

- *where* in the world you are studying
- what *stage* of your degree you are at, from first year through to final year
- what *level* you are studying at, foundation, undergraduate or postgraduate
- what *institution* you are studying at, this book is University and institution agnostic
- what *subject* you are studying, as long as you are **computationally curious**

There is something in this guidebook for *any* student of computing, both those inside and outside of Computer Science departments.

0.1 Imagining your future

This is a self-help guide but a lot of self-help literature can be dry, dull, text-booky, generic and boring with few illustrations and conversations. In the novel *Alice's Adventures in Wonderland* (Carroll, 1865) shown in figure 1, the protagonist Alice wonders why her sister is reading a book without pictures.

once or twice she had peeped into
the book her sister was reading,
but it had no pictures or
conversations in it, "and what is
the use of a book," thought Alice
"without pictures or
conversations?"

Alice's Adventures
in Wonderland



Figure 1: Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice "without pictures or conversations? (Carroll, 1865) Public domain image of the cover of the 1898 edition of the novel *Alice's Adventures in Wonderland* via Wikimedia Commons w.wiki/3S4C adapted using the Wikipedia app

Pictures tell stories, pictures explain. Pictures help you understand. Pictures help you imagine. So this book uses pictures (and conversations) to help you visualise your future. Other key differences between this and other guidebooks are outlined in section 1.12.

0.2 Your future aims

This guidebook aims to help you develop stronger habits of mind, body and soul using five key ingredients: **C, D, Y, F and .me**:

- 1. C is for CODE:** Instructions, algorithms, recipes, methods and strategies contained in this guidebook. This code is for your consumption, not for a machine.

2. **D is for DATA:** From big data to microdata, your data to my data. Structured data, un-structured data and semi-structured data. Facts, statistics, graphs, pictures and our data. Bytes, bits and bobs collected together for your analysis and amusement
3. **Y is for YOU:** This book is all about you, with activities and other *coding challenges* for you to do in addition to just passively reading
4. **F is for FUTURES:** Possible futures for you to think about. Try not to dwell on the past. Think about the future. Think about *your* future. (Ryder, 1988, 2019)
5. **.me is for ME:** Hello, my name is Duncan, see figure 2. I'm your tour guide here. If you're feeling a bit lost, follow me and together we can starting coding your future.



Figure 2: Hello, my name is Duncan. If you're feeling a bit lost, follow me and together we can starting coding your future. Please don't call me **Sir, Professor or Doctor**, just call me **Duncan**. Titles are tinsel! (Shelley, 1812) Image adapted from *Hello my name is ... sticker* by Eviatar Bach, public domain w.wiki/32RV

Coding your future explores techniques for investigating career possibilities, job searching, making career decisions, submitting applications and competing successfully in interviews and the workplace.

Alongside these practical engineering issues, this guidebook also encourages you to *design your future* by taking a step back and reflecting on the bigger picture. You will apply computational thinking techniques, to reflect on who you are, what your story is, how you communicate with other people and what your

experience is. As there is a computational theme, you will also need to reflect on what your inputs and outputs (I/O) are, both now and in the future. You'll also need to think about what recipes (or algorithms) you might start experimenting with.

This guidebook investigates professional and pastoral issues in computing, for those with and without Computer Science degrees in the early stage of their careers.

0.3 What you won't learn

This guidebook will NOT teach you how to write code, there's already lots of fantastic resources to help you do that. We discuss some of them in chapter 5 on *computing your future*.

0.4 Learning your future

So what *will* you learn from this guidebook? After reading this guidebook, watching the videos and doing the exercises you will be able to:

1. Improve your self-awareness by describing who you are, what motivates you and your strengths and weaknesses
2. Decide on and adjust your job search strategies
3. Identify employers, sectors and roles that are of interest to you
4. Improve your written communication skills both for job applications and communicating with other people
5. Plan and prepare competitive written applications using standard techniques including CVs, covering letters, application forms and digital profiles
6. Compete confidently and successfully in interviews. Anticipate and prepare for technical and non-technical questions
7. Plan further possibilities in your career such as promotion, postgraduate study & research, alternative employment and longer term goals
8. Search and navigate a large “wordbase” (this guidebook and the work it cites). A wordbase is like a **codebase**, only written predominantly in natural language.

0.4.1 Your future requirements

As the title of this guidebook implies, there is a computational flavour here, but you do not have to be studying Computer Science to benefit. There are two main target audiences for this guidebook:

1. Undergraduate and postgraduate students studying Computer Science as a major or minor part of their degree. This includes software engineering,

artificial intelligence, human-computer interaction (HCI), information systems, health informatics, data science, gaming, cybersecurity and all the other myriad flavours of Computer Science

2. Undergraduate and postgraduate students studying *any* subject, with little or no Computer Science at all. You are curious to know about what role computing could play in your future career because computing is too important to be left to Computer Scientists, see chapter 5 on *Computing your Future*

So the prerequisites for this book are that you are studying (or have studied) at University where English is one of the main spoken languages. You *may* have some experience already, either casual, voluntary or otherwise, but this book does **not** assume that you have already been employed in some capacity.

0.4.2 Gutting your future

Reading this book from cover to cover like a novel is not recommended. That would be foolish.

evisceration is the
removal of some or all of
the organs of the
gastrointestinal tract

Disembowelment



Figure 3: Don't *read* this book, disembowel it! Eviscerate it! Gut it like a fish! Enjoy the nourishing flesh and discard the less appetising organs of its gastrointestinal tract. You'll need to decide which is which, depending on your tastes and appetite. CC0 Public domain image of fish gutting by Wilfredor via Wikimedia commons w.wiki/_23m adapted using the Wikipedia app

Instead of reading this book, I suggest you follow the advice given to historian William Woodruff about reading books when he was at University:

“You don’t READ books, you GUT them!” — William Woodruff
(Woodruff, 2003)

So, gut this book like the fish in figure 3. Identify the chapters that are most useful to you (the flesh), and skip the rest (the guts). Which chapters are flesh and which are guts will depend on what stage of the journey you are at. This guidebook is designed to be as “guttable” as possible. To aid gutting, the version published at cdyf.me has a built in search and tables of contents. Before you can gut the fish, you’ll need an anatomical map shown in figure 4.

0.5 Mapping your future

This guidebook is split into four parts, shown in the map in figure 4:

1. In chapters 1 to 5 we look at *designing* your future
2. In chapters 6 to 13 we look at *building* and *testing* your future
3. In chapters 14 to 19 we look at *deploying* your future
4. In chapters 21 to 27 we hear from students who are *coding* their future

Let’s look in a bit more detail at the content of each of the four parts of this guidebook:

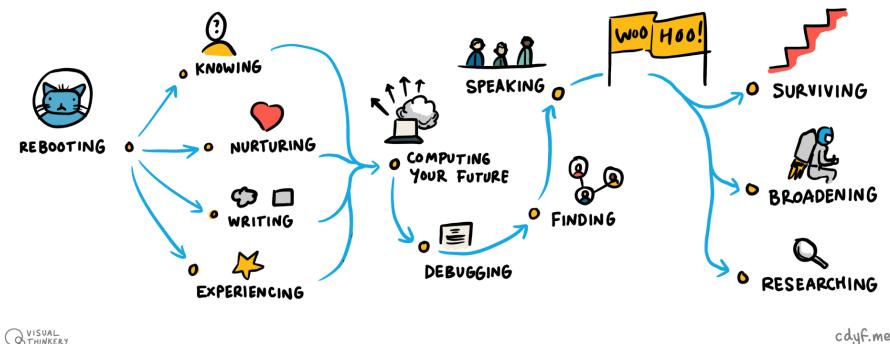


Figure 4: Mapping your future: Each yellow dot on this diagram is a chapter in *Coding Your Future*. The chapters on the left tackle design issues like *who are you?* Chapters on the right tackle the practicalities of executing and testing your career choices, such as *debugging your CV*. Mapping your Future artwork by Visual Thinkery is licenced under CC-BY-ND

0.5.1 Designing your future

The first six chapters of this guidebook look at what engineers call *design*. When you build anything, a bridge, a piece of software, a car or a plane you’ll need to do some design like the blueprint in figure 5

Building a career isn’t that different to building anything else, you’ll need to do some design work and it will probably be iterative. Designing things often

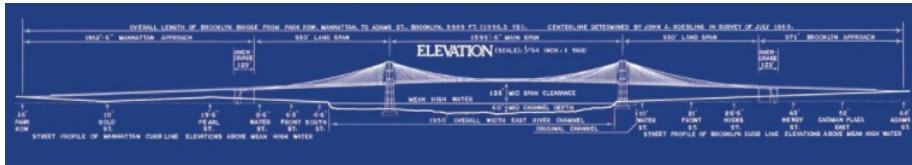


Figure 5: Designing your future is about drawing up a blueprint, like this one for the elevation of the Brooklyn Bridge in New York. What does your blueprint look like? Chapter's 1 through to 5 will help you design your future.

involves answering tricky questions. So when you're designing your future you'll need to cover the following:

- Chapter 1: *Rebooting your future* discusses why you should bother reading this guidebook
- Chapter 2: *Knowing your future* challenges you to reflect on who you are, what makes you unique and what you have to offer
- Chapter 3: *Nurturing your future* encourages you to pay attention to your mental and physical health
- Chapter 4: *Writing your future* explores your soft communication skills, how they complement your hard skills and why employers value them so much
- Chapter 5: *Computing your future* looks at the role computing can play in your career, especially if Computer Science is not a major part of your degree

0.5.2 Building and testing your future

The next seven chapters look at building (and testing) your future, what engineers like to call *implementation* or *execution* shown in figure 6.

Once you've started to answer the design questions in the first part, you can start to implement (or build) your career and think about what the next steps will be.

- Chapter 6: *Debugging your future* looks at debugging your written communication such as covering letters, application forms and digital portfolios.
- Chapter 7: *Experiencing your future* asks you to reflect on your experience and help identify where you can improve it
- Chapter 8: *Finding your future* looks at where and how can you look for interesting opportunities
- Chapter 9: *Broadening your future* encourages you to broaden your horizons. What are the possibilities beyond the obvious?
- Chapter 10: *Speaking your future* looks how can you turn interviews to your advantage and negotiate any offers you receive
- Chapter 11: *Surviving your future* looks at the next steps. Once you've



Figure 6: Just like the Manhattan Bridge, your future will be easier to build once you've done some preliminary design. You don't need a grand design with tonnes of details, a simple sketch will do. Design questions are covered in the first part of this guidebook on designing your future. Picture of the Manhattan bridge under construction in 1909 adapted from a public domain image via Wikimedia commons w.wiki/32Rg

- landed a job, how will you survive and thrive outside (and after) University
- Chapter 12: *Achieving your future* looks at evidence you can collect of your learning and development using various kinds of certifiable evidence
 - Chapter 13: *Researching your future* discusses if a Masters degree or a PhD right for you?

0.5.3 Deploying your future

The third part of this book, looks at deployment issues that follow from the design, build and test phases above. You'll need good deployment strategies to help with the inevitable stresses and strains of building your future as shown in 7



Figure 7: Huge supporting chains on the Clifton Suspension Bridge in Bristol allow heavy loads pass over the river Avon. You'll need good support to cope with the stresses and strains of building, testing and deploying your future. Clifton suspension bridge picture adapted from original by Nic Trott via Wikimedia commons w.wiki/32tu

- Chapter 14: *Ruling your future* provides *Ten Simple Rules for Coding your Future*, this book in a nutshell
- Chapter 15: *Hacking your future* invites you to put yourself in the employers shoes by hacking other people's CVs
- Chapter 16: *Moving your future* looks at opportunities outside of capital cities like London
- Chapter 17: *Enjoying your future* provides a soundtrack that might help with your wellbeing
- Chapter 18: *Actioning your future* gets you to think about your actions by emphasising verbs on your job applications
- Chapter 19: *Scheduling your future* is the live synchronous sessions for this course, if you're not participating in these, schedule a time every day or week for personal development

The final part, *Coding your Future*, from chapter 20 gives an overview of the podcast which accompanies this book: *Hearing your Future*. Each subsequent chapter meets a student who is *Coding their Future* and asks, how did they get to where they are and where are they going next? The final chapter 27: *Reading your future* lists everything cited in this guidebook.

0.6 Your future themes

This guidebook aims to help you build a bridge from where you are now to where you'd like to be in the future. Each chapter of the book contains the following recurring themes:



Figure 8: This guidebook will help you build a bridge to your future. Picture of the iconic Golden Gate Bridge in California during the blue hour adapted from an original by Frank Schulenburg (CC BY-SA) on Wikimedia Commons w.wiki/37kY

1. **Learning** your future: What you will learn from any given chapter
2. **Watching** your future: videos and animations for you to watch
3. **Listening** to your future: audio and podcasts for you to listen to
4. **Speaking** your future: articulating from a script or by improvisation, particularly when preparing for interviews
5. **Discussing** your future: breakpoints invite you to pause your code and think about the variables and parameters you are using. Can they be

- improved? Reflect and discuss.
6. **Reading** your future: because reading is good for your mind, body and soul. Read The Friendly Manual. RTFM. Read THIS Friendly Manual.
 7. **Writing** your future: written exercises using natural language
 8. **Quizzing** your future: quick quizzes to be done in real-time live scheduled sessions described in chapter 19 (synchronously) and in your own time (asynchronously)
 9. **Assessing** your future: activities to be assessed by yourself, your peers, an employer or an academic (depending on who and where you are)
 10. **Challenging** your future: coding challenges are designed to take you out of your comfort zone by encouraging you to experiment with your thoughts, discussions and actions
 11. **Signposting** your future: the most useful resources that I recommend you read, listen to or watch

0.7 Downloading your future

The full text of this guidebook is freely available at www.cdyf.me, this means the web version (that's all the *.html) is searchable, browsable and linkable in any web browser on your phone, tablet, laptop or desktop computer. If you'd prefer to read this guidebook in a single ebook file, you can download a copy at

- cdyf.pdf
- cdyf.epub

If you'd like to read this guidebook on your Kindle you can transfer the epub to your Kindle using amazon.com/gp/sendtokindle.

In the future, a traditional printed paper copy from a publisher may also be available. If you're a publisher who'd like to publish this book the old fashioned way, please get in touch.

0.8 Contributing to your future

If you'd like to contribute this guidebook, I welcome constructive feedback from loyal opposition and critical friends, see figure 9. All contributions will be gratefully acknowledged in section 0.9 unless you ask for your contributions to remain anonymous. If you're about to graduate or have already graduated in Computer Science, see section 20.7.

If you find what you're reading here useful and you think other people might benefit too, I'd really appreciate some stars (likes) on the guidebook's repository at github.com/dullhunk/cdyf to help other people find us.

I'm looking for feedback and contributions on everything in this guidebook from the small things like typos, grammatical errors and spelling mistakes through to bigger issues for each chapter such as:

A critical friend is a supportive person who can ask difficult questions using critical thinking to judge a situation

Critical friend



WIKIPEDIA

Made with the Wikipedia app

Figure 9: Can you be a supportive but critical friend of this guidebook? Public domain image of a painting *Friendship* by Petrona Viera via Wikimedia Commons w.wiki/3WjY adapted using the Wikipedia App

- Does the chapter make sense, is it clear?
- Does it strike the right tone, is it pitched at the right level? Not patronising? Too many platitudes?
- Are there too many motivational (or demotivational) quotations?
- Where is it too long and waffly (see figure 6.16) or too short?
- Are there too many (or too few) pictures? What needs more illustration?
- Is it well scoped? Too broad or too narrow?
- Are the stated learning objectives met by the chapter?
- Are the activities clear? Can students understand why the activities are recommended? What other activities could be added?
- Will it make sense to global readers e.g. will students from America, China and India etc understand the quirks and idioms of English language and culture
- Are there too many metaphors? Mixed metaphors? Awkward analogies? Idiotic idioms? Annoying alliterations?
- Too many citations? Not enough citations? Missed any key citations?
- What's missing?
- Where are the unstated assumptions? Where is the unconscious bias?
- What are the issues with equality, diversity and inclusion?
- Are there too many musical references or annoying emoji? Please bear in mind I'm trying to strike an irreverent, light-hearted and playful tone to improve readability
- What else should be ruthlessly edited out?

All suggestions welcome! Don't be shy. There are several ways you can contribute, depending on how comfortable you are with Git:

0.8.1 For git contributors

If you're familiar with `git` and `markdown`, there are several options if you have a github account (see github.com/join) including:

- Join the discussion at github.com/dullhunk/cdyf/discussions
- Raise new issues at github.com/dullhunk/cdyf/issues/new
- Click on the `Edit this page` link, which appears on the bottom right hand side of every page published at cdyf.me when viewed with a reasonably large screen (not a phone)
- Contribute at github.com/dullhunk/cdyf/contribute and help with existing issues at github.com/dullhunk/cdyf/issues
- Fork the repository, make changes and submit a pull request github.com/dullhunk/cdyf/pulls. If you need to brush-up on your pulling skills see makeapullrequest.com
- From the command line, clone the repository and submit pull requests from your own setup:

```
git clone https://github.com/dullhunk/cdyf.git
```

Most of the guidebook is generated from RMarkdown, that's all the `*.Rmd` files. So markdown files are the only ones you should edit because everything else is generated from them including the `*.html`, `*.tex`, `*.pdf`, `*.epub` and `*.docx` files.

0.8.2 For everyone else

If you don't want to (or can't) use `git` and github.com then you can:

- Add comments by annotating `cdyf.pdf` or `cdyf.epub` using your personal tablet of choice (iPad, reMarkable or whatever) and emailing your updated version to me
- Suggest changes by editing the Microsoft Word version at `cdyf.docx`. The text is all there, but the images are all over the place. This is because the typesetting algorithms in Word aren't anything like as good as the LaTeX ones used to create the `cdyf.pdf` (output) from the `cdyf.tex` (input).¹ Make sure you've turned on track changes in Word, one of it's killer features that allows your corrections to be easily identified from the original text.
- Just email me suggestions for improvements

Any corrections or suggestions will be gratefully received and noted in the acknowledgements section 0.9, unless you tell me otherwise. I welcome all improvements, big and small.

¹Don't say I didn't warn you!

0.9 Acknowledgements

The content of this book is based on hundreds of conversations I have had with students of (mostly) Computer Science, Mathematics, Physics and Engineering, since 2012. It is also based on conversations I've had with their employers too.

Coding Comment

This acknowledgements section is really looooooong because I try to practice what I preach about the importance of expressing gratitude, see section 3.4. It also serves as a live demonstration of a (public) gratitude journal. Expressing gratitude, publicly and privately, is a simple and proven technique for improving your mental health. It will also improve the mental health of the people who you thank, and strengthen the communities that you are part of, see 3.8.

If you want to get to the main content of this book you can skip this and go straight to chapter 1.

0.9.1 Thank you students

First and foremost, I would like to thank all the students who have helped with this book, both directly and indirectly see figure 10.



Figure 10: If I have seen further it is by standing on the shoulders of giants. (Newton, 1675) Public domain image of Orion carrying his servant Cedalion on his shoulders via Wikimedia Commons w.wiki/_zZ2E

So, if you have studied some flavour of Computer Science at the University of Manchester since 2012, there's a high probability you have contributed to this book. Thank you for having the courage to tell me your stories. Thank you for being ambitious, hard working, talented, fearless, creative, inspirational and

listening to me (sometimes). It has been my pleasure and privilege to work with you all.

I'd especially like to thank current and former industrial experience (IE) students who have completed a year in industry as part of their degree as well as those who have done summer internships, either as part of the Master of Engineering (MEng) program or otherwise, particularly (in alphabetical order) Matt Akerman, Sami Alabed, Teodora Balmos, Luke Beamish, Eirik Björnerstedt, Liam Breeze, Jingxuan Chen, Raluca Cruceru, Petia Davidova, Maximilian Gama, Mihail Ghinea, David Green, Lloyd Henning, Cristian Ilin, Călin Ilie, Sneha Kandane, Joshua Langley, Struan McDonough, Milen Orfeev, Jason Ozuzu, Alice Păcuraru, Carmen Prăxedes, Kristina Radinova, Tom Robinson, Teodora Stoleru, Peter Sutton, Kamil Synak, Boris Vasilev and Brian Yim Tam. In addition, the PASS leaders and facilitators, (PASS2-2021, PASS2-2020, PASS2-2019 etc), UniCSmcr.com, HackSoc, CSSoc and Manchester Ultimate Programming members have all been influential on the content of this book. I've learned heaps by manually trawling through thousands of your CVs too, so if you've shown me a copy of your CV, thanks! Chapter 6 on *Debugging your future* (self assessment) and chapter 15 on *Hacking your future* (peer assessment) are based on the most common bugs (or are they features?) I've seen in CVs.



Figure 11: Posing on the BBC Breakfast red sofa with the winning student team at the BBC / Barclays University Technology Challenge (UTC) in MediaCityUK, Salford, Greater Manchester.

So, thank you students for being studious.

0.9.2 Thank you employers

Thanks to all the organisations who have employed students from the Department of Computer Science as either summer interns, year long placements or graduates. A big chunk of this guidebook documents their experience of employers and their graduate recruitment programs.

Thanks to Niall Beard and Sharif Salah at Google for introducing me to Google's Technical Writing course in section 4.5.2. Writing is rewriting!

So, thanks employers for employing our students.

0.9.3 Thank you colleagues

I've also had significant support from colleagues in the Department of Computer Science (@csmcr), and many other parts of the University: (engineering, natural sciences, social sciences, biology, medicine and health etc) and support staff at the University of Manchester. (@UoMCareers, @alumniUoM, @OfficialUoM)

Thank you Carole Goble for building the community that supported me through postgraduate study. Thanks for creating the environment which this book was written in, especially the e-Science lab, Information Management Group (IMG), Software Sustainability Institute (software.ac.uk) and their spin-offs. Thanks for patiently re-teaching me how to write better by covering early drafts of my Masters thesis in red ink and less patiently (on subsequent revisions) swear words.

Thank you Steve Furber for playing guitar in our “boy band” Tuning Complete. All we've got is your bass guitar, three chords and the truth. (Howard, 1951; Dylan and Hewson, 1988)

Thank you Jim Miles for encouraging me to write a book shortly after you offered me a job. I thought you were joking (about the book) but it actually turned out to be another one of your great ideas. Thanks Jim.

I'd also like to thank the only three people in the whole world who've had the ~~misfortune~~ pleasure of reading all of my PhD thesis cover to cover; Robert Stevens, Anil Wipat and Steve Pettifer. I suspect it was as painful for you to read as it was for me to write it. Thanks Robert for your relentless patience and giving me a well timed, well aimed kick up the (proverbial) arse to write this book in the Midland Hotel, Manchester at the May ball.

So, thank you colleagues for being collegiate. You make the University of Manchester an enjoyable place to work.

0.9.3.1 Thanks to academic staff

Thanks to past and present academic colleagues (see figure 12), PhD students and academic staff at the University of Manchester (and elsewhere) who have contributed to this guidebook and the environment it was written in. We are bound together by the power of weak ties (section 8.2.5) alongside stronger forces and friendships.

They include (in alphabetical order): Muideen Ajagbe, Pinar Alper, Sophia Ananiadou, Mikel Egaña Aranguren, Constantinos Astreos, Terri Attwood, Sam Bail, Robin Baker, Richard Banach, Riza Batista-Navarro, Michael Bada, Niall Beard, Sean Bechhofer, Dick Benton, Casey Bergman, Hannah Berry, Lynne Bianchi, Ahmad Bilal, Rupert Blackstone, Stewart Blakeway, Petrut Bogdan, Caroline Bowsher, Linda Brackenbury, Andy Brass, Judy Brewer, Christian Brenninkmeijer, Andy Bridge, Andy Brown, James Brooks, Gavin Brown, Nick Brown, Mihai Bujanca, Bob Callow, Alex Casson, Lloyd Cawthorne, Zhongyan



Figure 12: Wearing silly hats and even sillier frocks for a graduation ceremony in the Whitworth Building, Manchester in 2013. From left to right Alex Walker, Tim Morris, John Latham, Graham Gough, Yours Truly, Sean Bechhofer, Andrea Schalk, Gavin Brown, Toby Howard, Robert Stevens, Simon Harper, Barry Cheetham, Norman Paton, Bijan Parsia, Caroline Jay, Allan Ramsay, Darren Lunn, Nick Filer, Markel Vigo and Ulrike Sattler. Picture by Toby Howard.

Chen, Oscar Corcho, Grant Campbell, Angelo Cangelosi, Peter Capon, Andy Carpenter, Nicola Carrier, Thomas Carroll, Barry Cheetham, Ke Chen, Sarah Clinch, Hannah Cobb, Mike Croucher, Laurence Cook, Ian Cottam, Brian Cox, Carmel Dickinson, Simone Di Cola, Dave De Roure, Paul Dobson, Clare Dixon, Janine Dixon, Danny Dresner, Nick Drummond, Ian Dunlop, Warwick Dunn, Dominic Duxbury, Doug Edwards, Sean R. Edwards, Iliada Eleftheriou, Anas Elhag, Suzanne Embury, Michael Emes, Roland Ennos, Harry Epton, Alvaro Fernandes, Jonathan Ferns, Michele Filannino, Nick Filer, Michael Fisher, Paul Fisher, R. W. Foster, Steve Furber, Andre Freitas, Aphrodite Galata, Matthew Gamble, Jim Garside, Kristian Garza, Freddie Gent, Chris Gilbert, Danielle George, Richard Giordano, Birte Glimm, Carole Goble, Antoon Goderis, Rafael Gonçalves, Roy Goodacre, Graham Gough, Anastasios Gounaris, Bernardo Cuenca Grau, Peter R. Green, Keith Gull, John Gurd, Luke Hakes, Robert Haines, Guy Hanke, Lucy Harris, Angel Harper, Simon Harper, Alison Harvey, Jonathan Heathcote, Alex Henderson, Martin Henery, Gareth Henshall, Andrew Horn, Farid Kahn, Chris Hardacre, Matthew Horridge, Ian Horrocks, Toby Howard, Roger Hubbold, Luigi Iannone, Jane Ilsley, Jules Irenege, Daniel Jameson, Caroline Jay, Mirantha Jayathilaka, Marianne Johnson, Huw Jones, Simon Jupp, Yevgeny Kazakov, John Keane, Douglas Kell, Catriona Kennedy, Rachel Kenyon, Chris Knight, Joshua Knowles, Dirk Koch, Nikolaos Konstantinou, Christos Kotselidis, Ioannis Kotsopoulos, Oliver Kutz, Alice Larkin, Peter Lammich, John Latham, Kung-Kiu Lau, Hugo Lefevre, Dave Lester, Peter Li, Zewen Liu, Phil Lord, Mikel Luján and Darren Lunn... (continued after the gratuitous picture break of figure 13)

... (continued) Matthew Makin, Nicolas Matentzoglu, Paul Mativenga, Erica McAlister, Mary McGee Wood, April McMahon, Merc and members of the



Figure 13: Masters and Mistresses of Science, part of the MSc Computer Science class of 2003. This is a bit like Where's Wally: can you find me in the photo? Unlike Wally I'm *not* wearing a red and white stripy jumper. Picture by Richard Giordano.

Manchester University Mountaineering Club (MUMC), Simon Merrywest, Eleni Mikroyannidi, Zahra Montazeri, Colin Morris, Norman Morrison, Georgina Moulton, Boris Motik, Christoforos Moutafis, Tingting Mu, Ettore Murabito, Mustafa Mustafa, Javier Navaridas, Kostas Nikolou, Aleksandra Nenadic, Goran Nenadic, Paul Nutter, Steve McDermott, Jock McNaught, Mary McGee-Wood, Pedro Mendes, Sarah Mohammad-Qureshi, Tim Morris, Jennifer O'Brien, Tim O'Brien, Steve Oliver, Pierre Olivier, Mario Ramirez Orihuella, Stuart Owen, Ali Owruk, Liam Panchaud, Pavlos Petoumenos, David Petrescu, Luis Plana, Colin Puleston, Ignazio Palmisano, Dario Panada, Michael Parkin, Bijan Parsia, Jon Parkinson, Norman Paton, Jeff Pepper, Steve Pettifer, Ian Pratt-Hartmann, Mark Quinn, Rishi Ramgolam, Allan Ramsay, Magnus Rattray, Alasdair Rawsthorne, Farshid Rayhan, Alan Rector, Giles Reger, Graham Riley, David Robertson, Jeremy Rodgers, Clare Roebuck, Mauricio Jacobo Romero, Nancy Rothwell, William Rowe, Oliver Rhodes, David Rydeheard, Graham Riley, Daniella Ryding, Ulrike Sattler, Ahmed Saeed, Pejman Saeghe, Rizos Sakellariou, Pedro Sampaio, Sandra Sampaio, John Sargeant, Andrea Schalk, Viktor Schlegel, Renate Schmidt, Baris Serhan, Jonathan Shapiro, Liz Sheffield, Lynn Sheppard, Bushra Sikander, Lemn Sissay, Vangelis Simeonidis, Kieran Smallbone, Alastair Smith, Stian Soiland-Reyes, Nikesh Solanki, Irena Spasic, David Spendlove, Laurence Stamford, Robert Stevens, Alan Stokes, Shoaib Sufi, Andrew Stewart, James Sumner, Neil Swainston, John H. Tallis, Paul Taplin, Federico Tavella, Chris Taylor, Tom Thomson, Dave Thorne, David Tolui, Tony Trinci, Dimitri Tsarkov, Daniele Turi, Fiona Velez-Colby, Jake Vasilakes, Laura Vasques, Delia Vazquez, Giles Velarde, Chiara Del Vescovo, Markel Vigo, Sam de Visser, Andrei Voronkov,

Niels Walet, Alex Walker, Louise Walker, Simon Watson, Nicholas Weise, Dieter Wiechart, Igor Wodiany, Katy Wolstencroft, Natalie Wood, Chris Wroe, Crystal Wu, Lisheng Wu, Terry Wyatt, Yifan Xu, Viktor Yarmolenko, Yeliz Yesilada, He Yu, Serafeim Zanikolas, Xiao-Jun Zeng, Jun Zhao, Liping Zhao, Ning Zhang and Evgeny Zolin.

“Look, everyone has a book inside of them... which is exactly where I think it should, in most cases, remain”

Christopher Hitchens

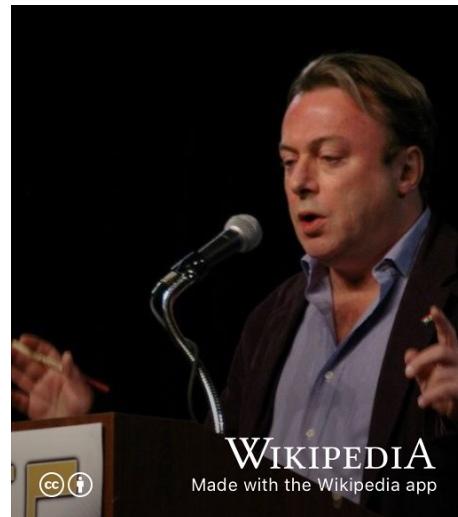


Figure 14: Optimists will tell you that “everyone has a book in them...”, but pessimists like Christopher Hitchens will add that “...in most cases that’s exactly where it should remain”. (Hitchens, 1997) Despite Hitchens amusing trademark scepticism, I am optimistic about the power of natural languages, written and spoken. CC BY portrait of Christopher Hitchens by ensceptico via Wikimedia Commons w.wiki/3YK7 adapted using the Wikipedia app

So thanks academics for being even more sceptical than Christopher Hitchens, see figure 14. Thanks academics for being academic.

0.9.3.2 Thank you professional services staff

Thanks also to the superb support staff (past and present) from professional services, especially the Academic Support Office (ACSO), Student Support Office (SSO) and external affairs office in the Kilburn building. Professional services staff continue to make all the magic of teaching and learning possible: Alyx Adams, Daniele Atkinson, Cassie Barlow, Jasmine Barrow, Jen-nie Ball-Foster, Emma Bentley, Christine Bowers, Ian Bradley, Karen Butterworth, Miriam Cadney, Chris Calland, Ben Carter, Chris Connolly, Han-nah Cousins, Amanda Conway, Ellie Crompton, Jean Davison, Holly Dews-nip, Gavin Donald, Kathryn Downey, Lindsay Dunn, Nicola Evans, Tammy Goldfeld, Penney Gordon-Lanes, Amelia Graham, Charlotte Hart, Iain Hart, Kath Hopkins, Sarah Howard, Lynn Howarth, Yvonne Hung, Susie Hymas,

Radina Ivanova, Dan Jagger, Alex Jones, Jeremy Jones, Jessicca Kateryniuk-Smith, Mike Keeley, Stephanie Lee, Dominic Laing, Gill Lester, Jez Lloyd, Ruth Maddocks, Cameron Macdonald, Tony McDonald, Karon Mee, Anne Milligan, Rachel Mutters, Matthew Oakley, Alyson Owens, Chris Page, Carly Peesapati, Melanie Price, Abby Ragazzon-Smith, Chris Rhodes, Stephen Rhodes, Graham Richardson, Martin Ross, Emily Sagues, Julian Skyrme, Elaine Sheehan, Angela Standish, Martine Storey, Bernard Strutt, Hannah Thomas, Jannine Thomas, Joseph Tirone, Daisy Towers, Karen Varty, Anna Warburton-Ball, Richard Ward, Sarah White, Elizabeth Wilkinson, Andrew Whitmore, Lisa Wright and Mabel Yau.

And Wendy. We all miss you and love you Wendy. #JusticeForWendy Fight the Power! (Ridenhour et al., 1989)

So, thanks professional services staff for being professional and supporting the work of academics doing research and teaching.

0.9.4 Thanks to funders

Thanks to support, financial and otherwise, at various stages from the following funding bodies:

- the Council of Professors and Heads of Computing (CPHC)
- the European Union (EU) and European Molecular Biology Laboratory (EMBL)
- the National Science Foundation (NSF)
- The Student Loans Company for financing my undergraduate degree
- UK Research and Innovation (UKRI) and its constituent parts:
 - the Biotechnology and Biological Sciences Research Council (BBSRC)
 - the Engineering and Physical Sciences Research Council (EPSRC)
 - the Natural Environment Research Council (NERC)
- The Wikimedia Foundation, Wikimedia UK and the Royal Society.
- Wiltshire County Council for my undergraduate maintenance grant

Money makes the world go round and has enabled me to teach, learn and do research. So thanks funders for opening your purse strings.

0.9.5 Thank you SIGCSE

Thanks to the sigcse.org, the Special Interest Group (SIG) on Computer Science Education (CSE), part of the Association for Computing Machinery (acm.org). Thanks to my fellow uki-sigcse.acm.org board members Steven Bradley, Janet Carter, Tom Crick, Quintin Cutts, Rosanne English, Sally Fincher, Samia Kamal, Joseph McGuire and Sally Smith for your help, support and advice, see figure 15

Thanks to all the SIGCSE journal clubbers including Brett Becker, Neil Brown,

**SIGCSE is the
Association for
Computing Machinery's
(ACM) Special Interest
Group on Computer
Science Education**

SIGCSE



WIKIPEDIA

Made with the Wikipedia app

Figure 15: Every year in January practitioners and researchers in computing education, both within Computer Science departments and elsewhere gather for Computing Education Practice (CEP) in Durham. Come and join our vibrant and thriving community! Picture of Durham Cathedral by Mattbuck via Wikimedia Commons w.wiki/4acc adapted using the Wikipedia app

Ceredig Cattanach-Chell, Katie Cunningham, James Davenport, Rodrigo Ferreira, Colin Johnson, Michael Kölling, Nicola Looker, Julia Markel, Jim Patterson, Sue Sentance, David Sutton, Moshe Vardi, Jane Waite, Pierre Weill-Tessier and Michel Wermelinger. Many of our journal club conversations have fed directly into the content of this guidebook.

Thanks to Sally Fincher and Janet Finlay whose report Computing Graduate Employability: Sharing Practice (Fincher and Finlay, 2016) has had a big influence on this guidebook.

So thanks SIGCSE for being special and interesting.

0.9.6 Thank you scientists

There is a wider community of scientists, engineers and scholars that have influenced this guidebook:

- Thanks to David Malan (@malan) for CS50 which is an inspiration to me and many others. (Malan, 2010, 2020, 2021) Thanks to Cristian Bodnar for inviting David to run CS50 in Manchester in 2017 which was a great introduction to David's work (Malan, 2017)
- Thanks to Laurie Santos (@lauriesantos), for *The Science of Well-being* (TSOWB) (Santos, 2022b) which was been a big influence on this book

had a gradual but significant effect on my personal and professional life. I've tried to distill some of the ideas into chapter 3 on *Nurturing your future*

- Thanks to Hadley Wickham (@hadley) and Garrett Grolemund (@garrettgman) for *R for Data Science* (Wickham and Grolemund, 2017) which helped me get started with R and bookdown. If you're reading this page in some kind of web browser, the stylesheet used here is re-used from r4ds.had.co.nz
- Thanks to David Walker for his book *Energy, Plants & Man* which inspired the conversations and pictures idea behind this book. (Walker and Walker, 1992)
- Thanks to Dave Cliff for your entertaining guest lectures for COMP101, see also figure 3.6

So thanks scientists (and engineers) for being scientific and engineering.

0.9.7 Thank you Bath

Thanks to the University of Bath for your excellent Postgraduate Certificate in Education (PGCE) course. I graduated with a PGCE in Science in 2011 and have been heavily influenced by the fantastic work of the schools in Swindon (section 0.9.9) and Shaftesbury (section 0.9.8) where I trained. I also learnt heaps from fellow students on the course and its course leaders:

- Caroline Padley, Physics
- Steve Cooper, Chemistry
- Malcolm Ingram, Biology

So thanks Bath for the initial teacher training (ITT), TeamBath™, the medicinal Aquae Sulis and the beautiful Cotswolds Area of Outstanding Natural Beauty (AONB).

0.9.8 Thank you Shaftesbury

Thanks to Chris Almond, David Ball, David Booth, Caroline Dallimore, Stuart Ferguson, Caroline Moss, Mr Travers and all the other staff and students at Shaftesbury School who hosted my first PGCE teaching placement, see figure 17. Thanks also to my fellow Shaftesbury and Bath trainees Katharine Platt, Harriet Edwards, Vicky Dury and Joan Shaw for sharing your knowledge through peer learning and peer instruction. Thanks Joan for keeping me awake on the long and winding west country roads to and from deepest darkest Dorset. Thanks for sharing the heavy burden of driving too.

So thanks Shaftesbury for lessons on top of Gold Hill and the Hovis Advert, one of Britain's best-loved adverts. (Scott, 1974)

Established in 1966, the University of Bath was named University of the Year by The Sunday Times in 2011

Bath, Somerset

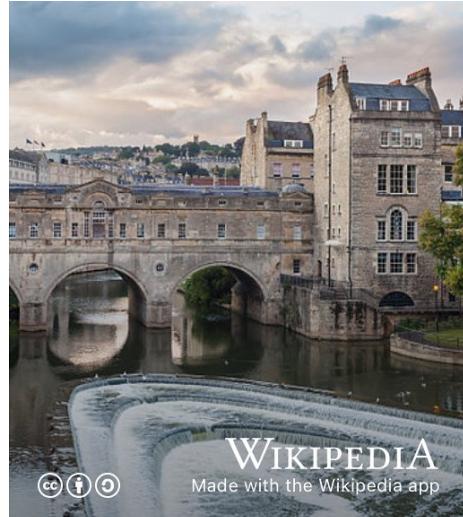


Figure 16: Named after its Roman Baths, the City of Bath is home to the University of Bath which was named *Sunday Times* University of the Year in 2011. Picture of Pulteney Bridge by Diego Delso, delso.photo, License CC-BY-SA via Wikimedia Commons at w.wiki/3VWY adapted using the Wikipedia app

0.9.9 Thank you Swindon

Thanks to headteacher & physicist Clive Zimmerman, his team of staff, Mr M. Carter , Mr K. Thomas and the students of Greendown Community School (now Lydiard Park Academy) in Swindon, Wiltshire for hosting my second PGCE teaching placement. It was fun teaching you about electromagnetic waves using Alom Shaha's jelly babies and kebab sticks shown in figure 18.

So thanks Swindon for being great and western and Swindon Town Football Club, the best football team in the whole of the West Country. Proper job.

0.9.10 Thank you Stockport

Thanks to headteacher Joanne Meredith, her team of staff and the students at St. Annes Roman Catholic High School, Stockport for hosting my Newly Qualified Teacher (NQT) year. Thanks to Rebecca Dann, Michael Doody, Keith Doran and other members of the alternative (Elizabethan) staff room for your emotional, moral and practical support throughout a challenging year fuelled by my midlife crisis. According to the *Manchester Evening News*, St. Anne's is "the forgotten school" (Johnson, 2020; Gill and Statham, 2021), see figure 19, but I'll never forget you or the lessons you taught me.

So thanks Stockport for being Stockport. Thanks for the magnificent Stockport

Gold Hill is a steep cobbled street in the town of Shaftesbury in the English county of Dorset. The view looking down from the top of the street has been described as "one of the most romantic sights in England."

Gold Hill, Shaftesbury



Figure 17: Shaftesbury in Dorset is the home of Gold Hill and Shaftesbury School. In the heart of https://en.wikipedia.org/wiki/Thomas_Hardy%27s_Wessex it has been used film location for several films including a screen adaptation of *Far from the Madding Crowd* (Hardy, 1874). Image of Gold Hill by Sean Davis via Wikimedia Commons w.wiki/3LhD adapted using the Wikipedia app.

Figure 18: Alom Shaha demonstrates his awesome wave machine. Physics and jelly babies, what's not to like? You can also watch the 4 minute video embedded in this figure at youtu.be/VE520z_ugcU (Shaha, 2014)

The school has gained the sobriquet 'The Forgotten School' and had nine head-teachers in ten years.

St Anne's RC
Voluntary Academy



Figure 19: Good governance is crucial to good schools. Many forgotten schools like St. Anne's R.C. High School, and the thousands of children in the UK they educate every year, need help from skilled people like you on their governing boards. Why not serve your local community as a “critical friend” on the governing board of a school? All ages are welcome, but especially younger governors, see where are all the young school governors? (Tickle, 2015) Take a look at governorsforschools.org.uk. Fair use image via Wikimedia Commons w.wiki/3Swt adapted using the Wikipedia app

Viaduct and for The Hatters: It's all that matters, Stockport Hatters.

0.9.11 Thank you schools

Thanks to all the schools who've hosted our undergraduate students as part of an ongoing partnership between the University of Manchester and local schools called Coding their Future:

- Mrs Rowland at Fairfield High School for Girls, Droylsden
- Mr Sinnott at Trinity Church of England High School, Hulme
- Mr Clarke at University Technical College (UTC@MediaCityUK), Salford
- Mr Jalloh at Manchester Communication Academy, Harpurhey
- Mrs Wood at The Barlow Roman Catholic High School, Didsbury
- Alan J. Harrison at William Hulme's Grammar School, Whalley Range
- Mr Rath and Mrs Preddy at Cheadle Hulme High School, Stockport
- Mrs Murray at Laurus Cheadle Hulme, Stockport
- Mr Pownall and Mr. Clarke at Knutsford Academy, Knutsford
- Steve Pearce at Altrincham Grammar School for Girls, Trafford
- Pauline Wilcox at Altrincham Grammar School for Boys, Trafford
- Mr Millington and Mr. Charlton at Manchester Grammar School (MGS), Fallowfield
- Mr Stenhouse at Stretford Grammar School, Trafford

Thanks to Mr Shaw for hosting our primary school codeclub.org. Thanks to Mr Ince and Drew Povey for showing me around Harrop Fold School (now The Lowry Academy) in Salford, host of the Educating Greater Manchester television series on Channel 4. (Sandwell, 2020)



Figure 20: Thanks to all the schools in Greater Manchester who've supported our Coding their Future partnership where undergraduates teach computer science in local secondary schools.

Thanks to all the schools who interviewed me for my Newly Qualified Teacher (NQT) year. Doing interview lessons, meeting your students and your senior leadership teams was a gruelling but fascinating magical mystery tour of the UK education system, both public and private. These interviews were very **productive failures**:

- Marie Getheridge at Writhlington School, Radstock, Somerset
 - see their amazing Orchid project wsbeorchids.org run by Simon Pugh-Jones
- The Cooper School, Bicester, Oxfordshire
- Patrick Hazlewood at St John's Marlborough, Wiltshire
 - not to be confused its posher and famous next door neighbour Marlborough College
- Oasis Academy, Brislington, Bristol
- Redland Green School, Redland, Bristol
- The John of Gaunt School, Trowbridge, Wiltshire
- Rachael Warwick at Didcot Girls' School, Didcot, Oxfordshire
- Vicky Tuck at Cheltenham Ladies' College, Cheltenham, Gloucestershire. (Anon, 2010; Wilby, 2011)
 - This was the first time I'd ever set foot in a private school, managing to cross Michael Gove's Berlin Wall of education by sneaking through Checkpoint Charlie un-challenged. It was an enlightening experience! (Gove, 2014)
- Blackburn College, Lancashire
 - “I read the news today, oh boy! Four thousand holes in Blackburn, Lancashire” (Lennon and McCartney, 1967)

So thanks schools, for all the excellent work you do educating people, whatever their background.

0.9.12 Thank you Oxford

Thanks to Martin Clutterbuck, Rebecca Clare, Richard O'Beirne, Simon Witter, Will Wilcox, Gavin, Howard, Isobel, Jess, Paddy, Sara, Spiro and everyone else in the journal production team at Blackwell Science Ltd for looking after me in my first job as a freshly minted graduate. Thanks to Nigel Blackwell, Bob Campbell and Jon Conibear without whom there wouldn't have been any Blackwell for me to Science at. Thanks to Tim, Ruth and Sarah for all the nights in Oxford pubs.

Thanks to Eileen, Anne & Richard for giving me a home from home.

Thanks to John Chelsom, Kal Ahmed, Clare Ashton, Tim Cave, Mavis Cournane, Eddie Dillon, Niki Dinsey, Phil Gooch, Antony Grinyer, Debbie Hagger, Gareth Hudson, Steve Horwood, Chris Joyce, Joe McCann, Eddie Moore, Keith McCann, Dave Nurse, Ian Packard, Mark Pengelly, Al Power, Lillian Spearing, Ron Summers, Omar Tamer and the rest of the team at (and clients of) CSW Informatics Ltd (csw.co.uk) for looking after me in my second job after Uni and teaching me about Oxford Innovation.

Thanks to my fellow xmlsummerschool.com faculty: Bob du Charme, Paul Downey, Michael Kay, Jeni Tennison, Norman Walsh and Lauren Wood for the memories and the <markup/>. (Johnson, 2004; Morali and Willis, 1978)



Figure 21: Looking West over Oxford's dreaming spires from South Park towards the city of Oxford. Picture adapted from an original by Tejvan Pettinger on Wikimedia Commons w.wiki/4Y25

```
X.S.L.T!
It's fun to program in... X.S.L.T!
Every line in your code
Is an XML node
And the program is one big tree
```

Thanks to Steven A. Hill, Jane Langdale and Chris Leaver at the University of Oxford (plants.ox.ac.uk) for interviewing me for a Gatsby Charitable Foundation DPhil scholarship. Thanks Chris for teaching me a painful but important lesson about the value of my education and grades.

So thanks Oxford for your dreaming spires, see figure 21.

0.9.13 Thank you Cambridge

Thanks to Christoph Steinbeck, Nico Adams, Marcus Ennis, Janna Hastings, Paula de Matos, Adriano Dekker, Kenneth Haug, Jo McEntyre, Pablo Moreno, Helen Parkinson, Mark Rijnbeek and Susanna-Assunta Sansone at the European Bioinformatics Institute (EBI, see figure 22) for looking after me during my time in Cambridge. Thanks to Rolf Apweiler, Michael Ashburner, Ewan Birney, Graham Cameron and Janet Thornton without whom there wouldn't have been an EBI for me to work at.

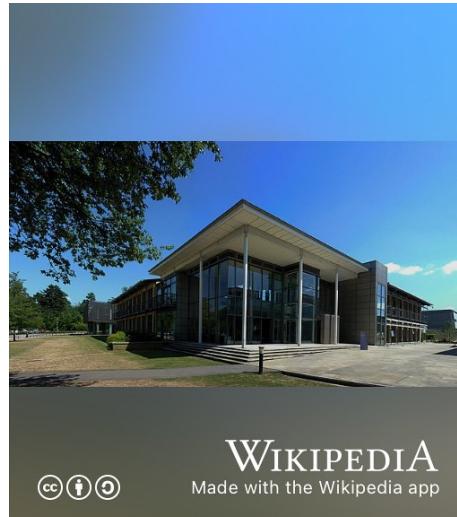
So thanks Cambridge for a really *fen-tastic* time in Silicon Fen.

0.9.14 Thank you Manchester

Thanks to Greater Mancunians beyond the University of Manchester: Anna, Mark Anderton, Andrea, Rob Aspin, Jon Atkinson, Charlie Ball, Paul Basson, Iain, Julian Bass, Amul Batra, Dean Belfield, Lisa Chan Brown, Martin Bryant, Gemma Cameron, Matthew Clark, Jeremy Coates, Craig, Darren Dancey, Craig Dean, Farhat Din, Anne Dornan, David Edmundson-Bird, Emily, Diana Erskine, Sherelle Fairweather, Shaun Fensom, Steven Flower, Tony Foggett, Katie Gallagher, Giles, Emma Grant, David Haikney, Damian Hughes, Mehran Jalaei, Daniel Jamieson, Matt Jarvis, Jamil Khalil, Ross Keep-

The European Bioinformatics Institute (EMBL-EBI) is an Intergovernmental Organization (IGO) which, as part of the European Molecular Biology Laboratory (EMBL) family, focuses on research and services in bioinformatics

European Bioinformatics Institute



WIKIPEDIA

Made with the Wikipedia app

Figure 22: The European Bioinformatics Institute (EBI) is an outstation of the European Molecular Biology Laboratory (EMBL) which carries out leading edge research and provides services in bioinformatics from Hinxton, just outside Cambridge, UK. Picture adapted from an original by Magnus Manske on Wikimedia Commons w.wiki/4YQB using the Wikipedia app

ing, Val Kelly, Kitty, David Levine, Julie Lowndes, Tony McGrath, Chris Marsh, Amy Mather, Lisa Mather, Claire McDonald, Keith Miller, Geraint North, Alan O'Donohoe, Tomas Paulik, Damian Payton, Francesco Petrogalli, Paul, Peppi, Phil, Rich, Ros, Miles Rothbury, Paul Sherwood, Howard Simms, Adrian Slatcher, Jason Souloglou, Joe Sparrow, Martyn Spink, Katie Steckles, Matt Squire, Julian Tait, Rob Taylor, Rachel Thompson, Tom, Andrew Toolan, Hannah Tracey, Wesley Verne, Paul Vlissidis, Tony Walsh, Travis Walton, Ben Webb, Paul Wilshaw and Zoe for friendly Northern support and advice. Thanks to Andrew Back and Tim Harbour for wuthering my bytes at wutheringbytes.com.

So thank you for the music the songs I'm singing. Thanks for all the joy they're bringing. Thanks Manchester for the best football team in the world and being Mancunian. This is the place! (see figure 24)

0.9.15 Thank you Coventry

Thanks to Phil Harris, Steph Harris, Alan Gear, Jackie Gear, Ally, Neil, Esther, Francis Rayns, Graham Smith, Jeremy Cherfas, Morgen Cheshire, Margi Lennartsson Turner, Lady Godiva (see figure 25) and everyone else at the Henry Doubleday Research Association (HDRA) and Coventry University for hosting my industrial experience year during my undergraduate degree.



Figure 23: Bees symbolise community and work ethic and have been a Manchester icon since the industrial revolution in the 19th Century. Bees also adorn our weekly *Wednesday Waggle* jobs newsletter for students waggle.cs.manchester.ac.uk. Buzzin'! Waggle dance artwork by Visual Thinkery is licensed under CC-BY-ND

Figure 24: I'm with Longfella (aka Tony Walsh) in believing that **THIS** is the place! You can watch the full 4 minute video embedded in this figure at youtu.be/PszMmYpQjPo (Longfella, 2017)

she rode naked – covered only in her long hair – through the streets of Coventry to gain a remission of the oppressive taxation that her husband, Leofric, imposed on his tenants

Lady Godiva

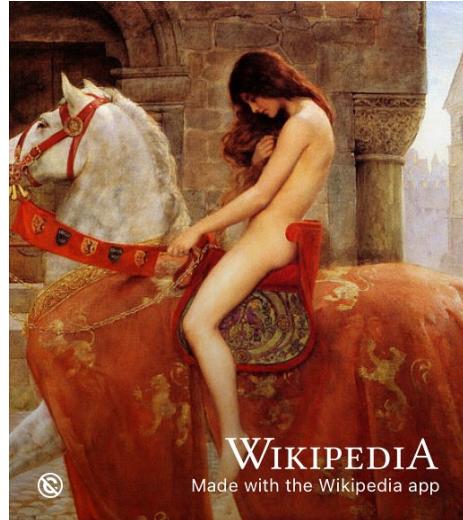


Figure 25: Covered only in her long hair, Lady Godiva rode naked through the streets of Coventry to protest about taxation. Sadly I was 900 years too late to miss the spectacle but there is a statue of her you can admire in Broadgate. Painting of Godiva by John Collier adapted from an original on Wikimedia Commons w.wiki/4aCU using the Wikipedia app

So thanks Coventry for naked women on horseback, a magnificent cathedral and the industrial experience. Thanks for being the place where I bumped into Bryan Mathers, see section 0.9.23.

0.9.16 Thank you Abisko

Thanks to Malcolm Press, Helena Björn van Praagh, Terry Callaghan, Jackie Potter, John Lee, Mats Sonesson, Nils-Åke Andersson, Rosie, Nick, Dylan, Karin, Kjell, Lennart, Marion, Martin, Ulf and everyone else at Abisko Scientific Research Station / Abisko Naturvetenskapliga Station (ANS, see figure 26) for hosting me as a summer research student investigating the effects climate change on subarctic heathlands. (Potter et al., 1995)² Easily the best summer job I've ever had!

So thanks (tack) Abisko for all the saunas, fika, midnight sun and Swedish hospitality.

0.9.17 Thank you America

Thanks to the British Universities North America Club (BUNAC) for sponsoring my Exchange Visitor Student Visa which allowed me spend an enjoyable summer

²You'll find me in the acknowledgements section of this paper

The Abisko Scientific Research Station (ANS) is a field research station managed by the Swedish Polar Research Secretariat. Situated on the south shore of Lake Torneträsk, it lies at the edge of the Abisko National Park

Abisko Scientific Research Station

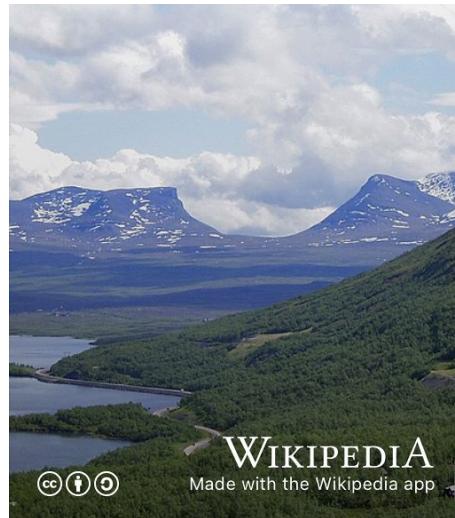


Figure 26: The Abisko Scientific Research Station (ANS) is a field research station in the Abisko National Park. The station hosts around 500 scientists each year from all over the world, who conduct research in subarctic environments. Picture of the view from Björkliden over the national park, past ANS on the shore of Lake Torneträsk towards the Gate to Lappland (Lapporten), by Lappländer on Wikimedia Commons w.wiki/4b3t adapted using the Wikipedia app

cooking breakfasts for guests at the Phillips Beach Plaza Hotel in Ocean City, Maryland. Thanks to Andy B. for flagging it.

BUNAC (British Universities North America Club) is an organisation that enables people to explore the world through work abroad and travel programmes

BUNAC

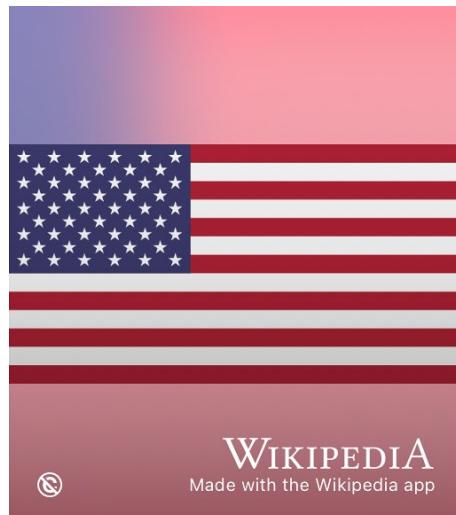


Figure 27: The British Universities North America Club (BUNAC) allows students to work abroad, see bunac.org. Public domain image of the flag of the United States via Wikimedia Commons w.wiki/4cRF adapted using the Wikipedia app

Thanks to Mitch at Green Tortoise Adventure Travel for driving, entertaining and feeding a bus load of us hippies with flowers in our hair from San Francisco to New York via Chicago and some of America's finest wildernesses (and cookouts) in Nevada, Utah, Wyoming, South Dakota, the Yellowstone, Grand Teton and Badlands National Parks. (Phillips and McKenzie, 1967)

Thanks to Tom and Letty Gochberg for your excellent hospitality in New York City, your transatlantic history lessons and showing me the very best that Manhattan has to offer. Thanks Pat, Colin and Rob Willmott for the introduction via the Single-handed Trans-Atlantic Race (STAR) in Plymouth, Devon from where the Mayflower (eventually) set off for the so-called New World in 1620, see figure 28.

Thanks Timo Hannay for letting me gatecrash the best party in Silicon Valley: Science Foo Camp (#scifoo) at the Googleplex in Mountain View, California in 2007 and again in 2009. Thanks to Cat Allman, Sergey Brin, Chris Di Bona, Tim O'Reilly and Larry Page for hosting scifoo. (Hull, 2007, 2009)

Thanks Boston, Massachusetts for the Pixies and hosting disruptive tea parties via the Sons of Liberty, the W3C Healthcare and Life Sciences Interest Group (HCLSIG) and the Association for the Advancement of Artificial Intelligence (AAAI) conference (Hull et al., 2006). Thanks Joanne Luciano for showing me

Figure 28: *Speedwell: No New Worlds* was an installation in Plymouth that invited the public to reflect on the legacy of the Mayflower’s journey, colonialism and the ecological state of our planet during the Mayflower 400 commemorations in 2020. The words remind us that while America may have been a “new world” to Europeans it had already been occupied by indigenous people for thousands of years. (Hampton et al., 2020) You can also watch the 4 minute video embedded in this figure at youtu.be/Y2Ya4d1oSQc (Hopes et al., 2020)

the sights of Cambridge, Massachusetts. Thanks to Ewa Deelman, Yolanda Gil and Bertram Ludäscher for hosting transatlantic workflow collaborations at the San Diego Supercomputer Center (UCSD) & University of Southern California (USC) with help from Carole Goble and funding from the Engineering and Physical Sciences Research Council (EPSRC).

So thanks America, I love you guys! Thanks America for being American.

0.9.18 Thank you Moravians

Thanks to Thespal Kundan, Principal of the Moravian Institute in Rajpur, Dehradun, Uttar Pradesh, India for hosting me and my friend Doug fresh out of high school on a gap year. We learned loads as visiting supply teachers of English and Mathematics, thanks to an introduction from a mutual contact Angus Barker, see figure 29.



Figure 29: The Moravian Institute lies in the foothills of the Himalayas between Dehradun in the Doon Valley and the hill station of Mussoorie. Situated between the Yamuna and Ganges, the institute was founded in 1963 by the late Reverend Eliyah Thsetsan Phuntsog in Ladakh, Jammu & Kashmir state to provide education for Tibetan refugees fleeing from their homeland across the Himalayas.

Thanks also to the Moravians in Manchester at Fairfield High School for Girls for hosting undergraduate Computer Science students as part of coding their future.

So thanks Moravians (and Angus) for life changing and formative experiences.

0.9.19 Thank you influencers

Some of the most important influences on this guidebook are people I've only met very briefly, virtually or not at all (yet).

- Thanks to Gayle Laakman McDowell (@gayle), for your cracking series of books (McDowell, 2014, 2015; McDowell and Bavaro, 2013; Bavaro and McDowell, 2021) which have been very useful resources both for students I've worked with and me personally

- Thanks to Yihui Xie (@yihui) and contributors to bookdown.org, the software used to produce this book alongwith the comprehensive and well-written documentation on using it. (Xie, 2017, 2015; Xie et al., 2020)
- Thanks to Bronnie Ware for your book *The Top Five Regrets of the Dying* (Ware, 2011) which helped me to re-align my values and priorities when they were all out of kilter
- Thanks to Jo Hobbs at Lancaster University for advice on placements and employability in undergraduate teaching

So, thanks influencers for being influential.

0.9.20 Thank you interwebs

Thanks to the artists, blaggers, bloggers, cartoonists, columnists, doodlers, diarists, essayists, film-makers, journalists, photographers, podcasters and writers whose words and pictures I've enjoyed reading, watching and listening to via the magic of the interwebs, see figure 30.



Figure 30: The pen is mightier than the sword, and so is the keyboard. Thank you mighty writers for your incisive words. Your writing provides an existence proof that everyone benefits from good communication. CC BY SA picture of a backlit keyboard adapted from an original by Colin on Wikimedia Commons [w.wiki/4sQd](https://commons.wikimedia.org/wiki/File:Backlit_keyboard.jpg)

So here are some people whose stuff I read, watch, listen to or use, maybe you'll enjoy their words, pictures and software too:

- Euan Adie at blog.overton.io/author/euan-adie
- Steven Appleby at twitter.com/stevenappleby, see figure 31
- Jonathan Black at ft.com/dear-jonathan
- Tim Bray at ongoing
- Geoffrey Challen at geoffreychallen.com/essays
- Jorge Cham at phdcomics.com
- Paul Downey at whatfettle.com and flickr.com/photos/psd
- Mike Croucher at walkingrandomly.com
- Stephen Curry at occamsotypewriter.org/scurry
- Athene Donald at occamsotypewriter.org/athenedonald

- Stephen Dubner at freakonomics.com
- Alf Eaton at hublog.hubmed.org
- Jonathan Eisen at phylogenomics.blogspot.com
- Michael Eisen at micaeleisen.org/blog
- Julia Evans at jvns.ca
- Martin Fenner at blog.front-matter.io
- Kevin Fong at bbc.in/35jDOwI
- Timothy Gowers at gowers.wordpress.com
- Mark Guzdial at computinged.wordpress.com
- Paul Graham at paulgraham.com
- Bosco Ho at boscoh.com
- Pierre Lindenbaum at github.com/lindenb
- Andrew Maynard at andrewmaynard.net
- Randall Munroe at xkcd.com
- Cameron Neylon at cameronneylon.net
- Peter Norvig at norvig.com
- Samuel Pepys at pepysdiary.com
- Neil Saunders because what he's doing is rather desperate
- Sue Sentance at suesentance.net/blog
- Kristin Stephens-Martinez at csedpodcast.org
- Greg Tyrelle at nodalpoint.org
- Greg Wilson at third-bit.com

So, thanks writers for writing. Thanks for penning, drawing and recording stuff that has informed, entertained and inspired me.

0.9.21 Thank you githubbers

Thanks to everyone who has contributed via github, listed below in order of github usernames. I will credit *any* github contributors here, small or large. Even the typos, it all counts. I don't care what operating system you are using either, see figure 32. You can easily add yourself to this roll call (see section 0.8) by correcting my delibreate mitsakes.

Aman (@amanrana1), Keith Mitchell (@apiadventures), Zee Somji (@ezeethg), iliketohelp (@iliketohelp), Jan Machacek (@janm399), teobalmos (@teobalmos), Tsvetankov (@Tsvetankov), Richard Gourley (@richardgourley), Tristan Maat (@TLATER), Safder Iqbal (@safderiqbal)

So, thanks githubbers for cloning, forking, merging, pulling, adding, committing and pushing.

0.9.22 Thank you Wikipedians

Thanks to all the thousands of editors and engineers that make Wikipedia one of the greatest communities on the internet, see figure 33.

Special wiki-thanks to English speaking Wikipedians Evan Amos, Abd Alsattar

Figure 31: Artist Steven Appleby explains some simple techniques for drawing. Useful techniques that can also be applied to writing anything, playing music, living life etc. The 6 minute video embedded in this figure can also be viewed at youtu.be/G_Vz7gTh1VE (Appleby, 2008)



Figure 32: Windows users meet in the office, Mac users meet in Starbucks while Linux users meet on github. Comic by Christiann MacAuley at sticky comics stickycomics.com/where-did-you-meet used with permission see stickycomics.com/permissions. Thanks Christiann.

Individual contributors are known as "Wikipedians". Anyone—including you—can become a Wikipedian by boldly making changes when they find something that can be improved

Wikipedia community



Figure 33: A small fraction of the Wikipedia community that works to give free access to the sum of all human knowledge to every single person on the planet. CC BY-SA picture of Wikipedians gathered at the annual Wikimania conference in 2012, adapted from an original by Helpameout on Wikimedia Commons [w.wiki/3YLJ](https://commons.wikimedia.org/w/index.php?title=File:Wikimania_2012_group_picture.jpg&oldid=3131110) using the Wikipedia app. Like this image, many of the illustrations in this guidebook are re-used or adapted from openly licensed images taken from commons.wikimedia.org

Ardati, Caroline Ball, Marianne Bamkin, Roger Bamford, Alex Bateman, Dan Brickley, John Byrne, Manu Cornet, Lucy Crompton-Reid, Daria Cybulska, Andrew Davidson, Paul Gardner, Madeleine Goodall, Aaron Halfaker, Melissa Highton, Eoin Houston, Dariusz Jemielniak, Chris Koerner, Darren Logan, Magnus Manske, Andy Mabbett, Charles Matthews, Ewan McAndrew, Daniel Metchen, Josh Minor, Peter Murray-Rust, Richard Nevell, Frank Norman, Paul Nurse, Rod Page, Bhavesh Patel, Mike Peel, Martin Poulter, Joseph Reagle, Frank Schulenburg, Gage Skidmore, Dario Taraborelli, Sara Thomas, Denny Vrandečić, Ian Watt, Alice White, Jessica Wade, Taha Yasseri for insights, inspiration, support, software, data, pictures and guidance. Thanks also for educating me on issues of equality, diversity and inclusion, especially gender and race.

So, thanks Wikipedians for being Wikipedia.

0.9.23 Thank you Bryan

Many of the illustrations for this book have been drawn by the very talented Bryan Mathers @BryanMMathers shown in figure 34.

Bryan is an artist, visual thinker, entrepreneur and listener who turns stories into pictures. He also happens to have a Bachelors degree in Computer Science from the University of Glasgow. As a renaissance man, his combined skills in art, science and engineering made him the perfect fit for illustrating this guidebook. You can find out more about Bryan at bryanmathers.com and visualthinkery.com. I'm very glad we randomly bumped into each other at the #wikiedu20 conference: wikiedusummit.coventry.domains.

So, thanks Bryan for your witty illustrations, this book wouldn't be the same without your visual thinkery.

0.9.24 Thank you St Laurence

Thanks to St Laurence comprehensive school (st-laurence.com), a community I am proud, lucky, privileged and grateful to have grown up in and still be part of decades later: Adam, Alan, Andrea, Andrew, Anna, Bouncing Barney, Charlotte, Catherine, Clare, Dan, Doug, Debbie, James, Jenny, Jim, Jo, John, Jon, Lou, Marcus, Marjorie, Matthew, Philip, Portia, Richard, Sasha, Scott, Simon, Sophie, Sophia, Stephen, Steve & Wilf. I'm especially grateful for the friendship of former St Laurence school students I've enjoyed music, cycling, football, walking, travelling, holidaying, drinking and camaraderie with, see figure 35. So:

- *Vive le Tour*
- *vive la différence*
- *Vive la Révolution*

I'm looking forward to the next revolution of our ongoing adventures.

Figure 34: People tell stories and stories paint pictures. Bryan Mathers, who has illustrated much of this guidebook, telling stories at TEDxGalway in 2021. You can watch the full 15 minute video embedded in this figure at youtu.be/IapGM5ZYBEw (Mathers, 2021)



Figure 35: Crossing the highest paved mountain pass in Europe, the Col de l'Iseran in France, with my fellow alpinists and Kings of the Mountains: Jim, Dan, Doug and Dan. Allez, allez, allez! Unlike professional riders in *Le Tour* we had heavy panniers and no performance enhancing drugs besides *vin de table* and *l'hospitalité française*. Rest in Peace Dan, you were a cherished friend, we all loved you and miss you terribly. Repose en paix.

Special thanks to former St Laurence school student and current sixth form head Aidan Blowers for showing me around The Clarendon School in Trowbridge, Wiltshire and leading by example. Aidan's performance as *Lord of the Dance* (wearing a white shirt in figure 36), inspired the ongoing musical experiment that is Tuning Complete.

Thanks to all my teachers at St Laurence school, some of whom can be seen in figure 37.

Thanks to the rest of my St Laurence school teachers not pictured in figure 37. In alphabetical order: Phil Arthur, Sally Arthur, Maggie Bignell, Jackie Bolton, Tony Brooks, Dave Brush, Mrs. Buthlay, Andrew Butterworth, Cathy Cooper, Ed Corrin, Mrs Davies, Brian Ellis, Myra Ettridge, Sue Glanville, Ms. Gledhill, Roger Greenwood, Barry Hales, Amanda Hodges, Steven Hollas, Maddy James, Mr Jones, Madame Lindsay, Karen Long, Sheila Macdonald, Simon Mitchell, Lee Musselwhite, Tim Noble, Roger Norgrove, Dave Pegg, Angela Pendennis, Sally Rose, Brian Reynolds, Steve Stretch, Mr Sadler, Mike Sullivan, Phil Smith, Rob Townhill, Beryl Tucker, Chris Watters, James Wetz and Bill Wheeler.³

Thanks to all of the school governors for holding the leaders of St. Laurence to account. I'm a #StateSchoolProud member of the 93percent.club.

So, thank you to the community that is St. Laurence School, for enabling my GCSE's followed by A-levels in Physics, Chemistry, Biology and General Studies.

³No, we didn't address teachers by their first names, it was always Mr X, Mrs Y or Miss Z but a bit of investigation has revealed their full identities.



Figure 36: Year 11 leavers of Melksham Oak Community School (MOCS) in Wiltshire dance to Uptown Funk with help from Mark Ronson, Bruno Mars and Aidan Blowers. The image in the figure is a screenshot. Don't believe me, just watch, come on! youtu.be/z8qH05teRMM (Ronson and Mars, 2014; Blowers, 2015)

Thanks for educating me (and many others) the West Country way. Proper job!
(Stoke and Green, 2013)

0.9.25 Thank you Fitzmaurice

Thanks to my Fitzmaurice Primary School school teachers: Mrs Cripps, Miss Clarry, Morris Clay, Neil Fleming, Mr Jackson, Betty Knowles, Mr. Lucas, Valerie Payne, Miss Sheldon, Hugh Solomon, Miss Uncles and Mrs White. I used to foolishly think it was secondary schools that did all the serious teaching, but they'd be *nowhere* without the crucial foundations laid in primary school. It takes a whole community (a village) to raise a child and a lot of that starts in primary school, see figure 39.

So thanks Fitzmaurice Primary School and Edmond Fitzmaurice for laying solid foundations.

0.9.26 Thank you Branwen

Thanks Branwen Munn, pictured on the left in figure 40, for introducing me (pictured on the right) to `while` loops on your very own Commodore 64 while playing Duran Duran IIRC. It doesn't get much more 1980s than that! Thanks to Mr. Jackson at Fitzmaurice Primary School for hosting an after school code club on the school's (one and only) computer, a BBC Micro. Like Emma Mulqueeny



Figure 37: The staff of Fitzmaurice Grammar School shortly before it merged with Trinity secondary modern school to form the comprehensive St Laurence school in 1980. Back row, left to right, Alistair Thomson, Tony Hull, Geoff Swift, Peter Knight, John Warburton, John Blowers, Stuart Ferguson, Tim Wilbur, Bob Hawkes, Harry Haddon, John Blake. Centre row: Joan Davis, Lynne Powell, Doug Anderson, Colin Steele, Virginia Evans, Joan Van Ryssen, Margaret Osbourne, Mireille (French Assistant), Sally Burden, Margaret Gadd. Front row: Ken Revill, Marilyn Maundrell, Noreen Brady, Sid Johnson, Gerald Reid (Headmaster), Meg Tottle-Smith, Enid Wicheard, Diane Satterthwaite, Liz Buchanan, Margaret Hore. Picture via Keith Berry. (Berry, 1998)



Figure 38: The badge of St. Laurence School. The emblem is an adaptation of the Fitzmaurice badge with a pair of gudgeon fish symbolising the union of the two schools St. Laurence was created from: Trinity secondary modern school and Fitzmaurice Grammar School.

"It takes a village to raise a child"

Hillary Clinton



Figure 39: It takes a village to raise a child. (Clinton, 1996) Bradford-on-Avon in Wiltshire is my village. Portrait of Hillary Clinton speaking in 2016 by Gage Skidmore on Wikimedia Commons w.wiki/4Xrc adapted using the Wikipedia app

I think year 8 is too late. (Mulqueeny, 2011) It is unlikely I would have ended up where I am if it hadn't been for early exposure to computing in primary school. (Danton, 2021)

So thanks Branwen, for all the breakdancing, birdwatching, music and computing.

0.9.27 Thank you NHS

Thank you NHS for all the healthcare you've provided for me and my family from our cradles to our graves, see figure 41. I had taken your services for granted until I thought I was on my deathbed, see section 2.2.7. Thankfully, I was granted some extra "Fergie time" from the Grim Reaper. (Pritchard, 2012)

So, thanks NHS for all the publicly funded healthcare.

0.9.28 Thank you family

To my family: wife, son, mum, dad, brother, sister, , and extended family: I'm lucky to have been taught by you and that you've always been there when I needed you.

So, thanks to all my family scattered around the world for your unconditional love.



Figure 40: Birdwatching at RSPB Radipole Lake in Weymouth, Dorset. Besides being active members of the Young Ornithologists' Club (YOC) we also enjoyed breakdancing, music and computing. Yes we are geeky and we'll always wear our geek badges with **pride!** Blessed are the geeks, see section 1.6.



Figure 41: Thank you NHS started in during the COVID-19 pandemic when people in the UK posted messages of gratitude for their National Health Service (NHS), to acknowledge their crucial work. Picture adapted from an original CC-BY-SA picture of the NHS logo painted on road outside North Walsham Hospital in Norfolk, England by Whippetsgalore on Wikimedia Commons [w.wiki/55Wc](https://commons.wikimedia.org/w/index.php?curid=55Wc)



Figure 42: The Acropolis of Athens is home to the Parthenon, a temple dedicated to Αthena : the goddess of wisdom and warfare. Picture of the Acropolis at night from the Pnyx with Hymettus in the background by George E. Koronaios adapted from an original on Wikimedia Commons w.wiki/4c2t.

Γ .

0.10 About me

Hello, my name is Duncan Hull and I'm currently writing⁴ this guidebook for undergraduate and postgraduate students as part of my day job at the University of Manchester where I'm a lecturer (Assistant Professor) in the Department of Computer Science.

So what's *my* story? I've been gainfully employed as a paperboy, supermarket cashier, shelf stacker, sausage packer, computer hacker, pork pie filler, plongeur, chef, dogsbody, field assistant, database administrator, deli counter server, consultant, matchday steward, envelope stuffer, high school teacher, postdoc, research scientist, chairperson, software engineer, lecturer, external examiner, tutor and scholar. Like many people, my path has been a bit of an *Odyssey* or what Helen Tupper and Sarah Ellis call a "squiggly career". (Tupper and Ellis, 2020) It's highly likely that, like me, your career will not follow a neat linear trajectory either. (Tupper and Ellis, 2021)

Beyond the paid stuff, I've done a range of voluntary work too, serving as a competition judge, fundraiser, code club & coderdojo leader, rabble rouser, digital council member, school governor, curator, librarian, beer drinker, wikipedia trainer, journal clubber and editor. But as Ronnie Lane and Ronnie Wood (figure 43) once said, I wish that I knew what I know now, when I was younger.

This guidebook documents some of what I know now, that I wish I'd known, when I was younger. If you're starting your career, I hope you find these insights and exercises useful. I've sat on both sides of many interview tables, as interviewer and interviewee. I have had some spectacular failures, alongside some modest successes, and have included personal stories where they are relevant.

⁴As you can probably tell, it isn't finished yet

"I wish that I knew what I know now, when I was younger."

Ooh La La (Faces song)



Figure 43: Hindsight is a great teacher. Poor old grandad I laughed at all his words but I wish that I knew what I know now, when I was younger, see Ooh La La (Lane and Wood, 1973) I've written some of what I know now in this guidebook, I hope you find it useful.

Most of what I have learned about employment comes from listening to, watching and helping students as they interact with employers on the first tentative steps in their careers, particularly through our industrial experience program, see figure 44.

I've documented some of what our students have taught me, so reading this book may help you learn from some of their successes and failures.

0.11 Legal stuff

I am not a lawyer (IANAL) but any opinions expressed in this guidebook are my own and not representative of my current employer, the University of Manchester. This guidebook does **NOT** therefore, represent University policy.

0.11.1 Licensing

The *text* of this guidebook is published under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License (CC-BY-NC-ND) license see figure 45.

This means you can copy and redistribute the written material provided that:

- You provide full attribution by linking directly to the original source
- You do not use the material for commercial purposes

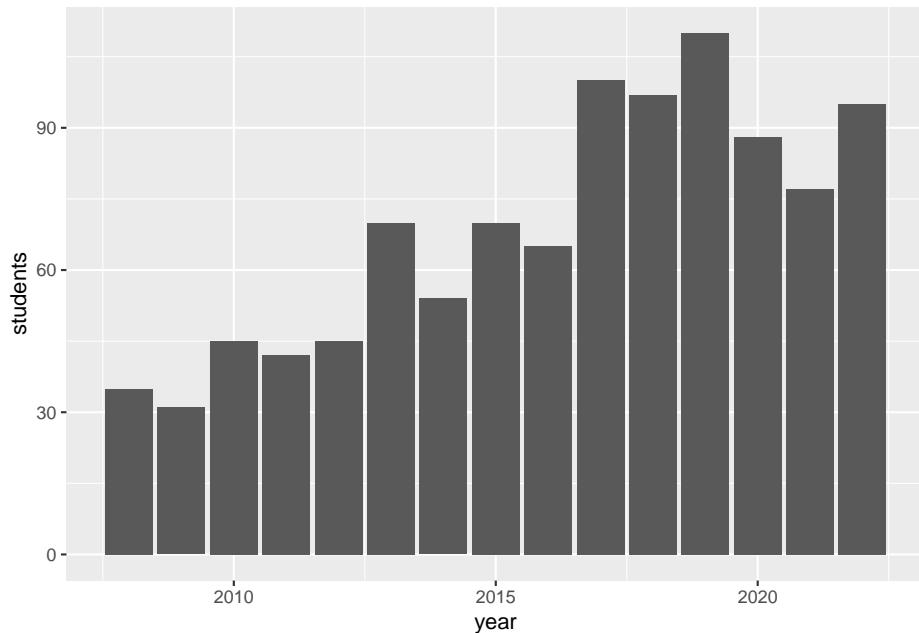


Figure 44: Number of undergraduate Computer Science students completing a year in industry as part of their degree at the University of Manchester. Since 2008 over 900 students have completed the program that I have been leading since 2012. We've managed to double the number of students doing placements per year. As you can see in the histogram, the COVID-19 pandemic started having an effect on students starting placements in 2020.



Figure 45: The *text* of this guidebook is published under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License (CC-BY-NC-ND) license which means you can copy and redistribute the material provided that you provide full attribution, do not use the material for commercial purposes and you do not make any derivative works.

- You do not make any derivative works

See the full license (CC-BY-NC-ND) for details.

The *images* used in this guidebook are published under different licenses, depending on their source. For example, Bryan Mathers illustrations are licensed CC-BY-ND, see figure 46. Other images have different licences, for example, images from Wikimedia Commons (commons.wikimedia.org) are published under CC-BY or CC-BY-SA, fair use or public domain. Each figure caption gives details for that images licence.

0.11.2 Your privacy

This site is hosted on netlify.com, see the netlify privacy policy. This site also uses netlify analytics (server side) and Google Analytics (client side) to understand our audience better. Both comply with the General Data Protection Regulation (GDPR). If you want to, you can opt out using the Google Analytics Opt-out Browser Add-on.

Some of these services use cookies. These can be disabled in your browser, see [allaboutcookies.org/manage-cookies](https://www.allaboutcookies.org/manage-cookies)

So now that we've dispensed with the formalities, let's look at why should you bother reading this guidebook in the first place.

CREATIVE COMMONS ATTRIBUTION

IF YOU'D LIKE TO USE SOME ARTWORK FOR YOUR
BLOGPOSTS, PRESENTATIONS & WEBSITES, HERE'S HOW!



KER POW by Bryan Mathers is licenced under CC-BY-ND 4.0



@bryanmathers
CC-BY-ND

Figure 46: The *images* in this guidebook are published under different licences, see each figures caption for details. Bryan Mathers illustrations are licensed CC-BY-ND, which means you should link to the original artwork, the creator profile and the licence terms. CC attribution artwork by Visual Thinkery is licensed under CC-BY-ND

Part I

DESIGNING

Chapter 1

Rebooting your future

The first part of this book is about *designing* your future. Before we get started, we need to reboot and tackle a fundamental design issue. Why the hell would you want to bother reading this guidebook when you have so many other things to do right now?

- You are a busy person, YES!
- Your time is a precious and finite resource, YES!
- You could be spending that precious time right now in lots of other ways, YES!
- There are mountains of self-help guides and courses already, YES!
- Do you really need *yet another* guidebook? YES! YES! YES!

You should read this guidebook because it is different to all the other guidebooks! It will help you design, test, build, debug and **code** your future in computing.

Before you start coding, we need to reboot. Come with me down the rabbit hole in figure 1.1 and let me explain...

1.1 What you will learn

After reading this chapter you will be able to reboot your future by :

1. Setting your expectations for using this guidebook, and open some doors to your future
2. Travelling down the rabbit hole into the underworld of employment
3. Discussing some of the gaps that exist between formal education and employment and how you can bridge them



Figure 1.1: Shall we go down the rabbit hole? Rabbit Hole learning by Visual Thinkery is licensed under CC-BY-ND

1.2 Let's go down the rabbit hole

In the novel *Alice's Adventures in Wonderland* (Carroll, 1865), the protagonist Alice follows a white rabbit down a hole. What she discovers is a strange underground world populated by weird and wonderful characters. The world of work can sometimes be a mysterious underworld where you adventure in wonderland accompanied by colourful characters.

You will spend lots of time in this wonderland, potentially as much as 80,000 hours of your life. (Investor, 2019, 2020) So join me down the rabbit hole, it's fun (honest), and sooner or later you'll have to come down here anyway. So open up the door to the new possibilities in your future.

1.3 Opening your future

Studying at University opens new doors to your future, some of which will take you down rabbit holes. As the poet Lemn Sissay puts it (figure 1.2):

**Open all doors, open all
senses
Open all defences, ask:
What were these closed
for?**

Lemn Sissay

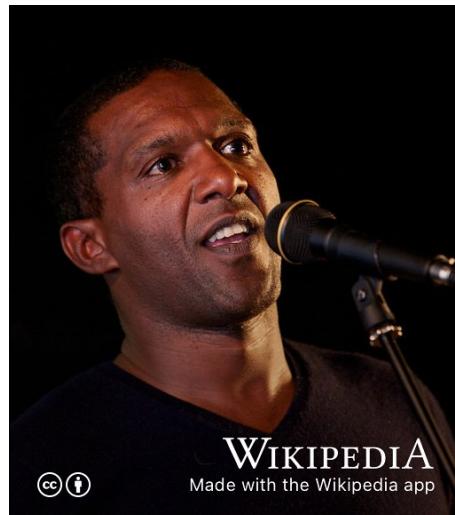


Figure 1.2: Open all doors, open all senses, open all defences, ask: What were these closed for? From *Inspire and be Inspired* by Lemn Sissay whose poetry is even better when you hear it, rather than just read it youtu.be/WzZs1w3NWzg. (Sissay, 2015) Portrait of Sissay speaking in 2010 by Philosophy Football via Wikimedia Commons w.wiki/3VYT adapted using the Wikipedia app

So University is a fantastic opportunity for you to open lots of new doors in your life and see where they take you.

1.4 Maximising your future

As well as opening your future, studying at University is about *investing* in your future. You're spending lots of your time and money at University. Hopefully, you've picked a subject that stimulates and challenges you intellectually while allowing you to find and develop your unique talents. But there's another reason that you probably chose to study at University and that was to improve your job prospects. This guidebook will:

1. Help you maximise the return on investment (ROI) of the substantial amount of your time and money you have already put into your education, from high school through to University
2. Give you an overview of important professional issues that are sometimes neglected or sidelined in school and University curricula
3. Highlight and review essential resources beyond this guidebook that will help with the above

All of the resources that can help you are scattered around in lots of different places. There are books, there are videos, there are podcasts, there are websites and jobs boards. There are online courses, blogs, social media, newspaper columns, journal articles, marketing material and many other good resources. It is overwhelming.

1.5 Your future is your responsibility

When Andy Stanford-Clark started working at IBM as a graduate fresh out of University, his manager gave him the career advice shown in figure 1.3:

Andy is now Chief Technology Officer (CTO) of IBM in the UK and an IBM Master inventor, so it was probably good advice. There are plenty people who can help design and build your future, ultimately it is **YOU** who has to take responsibility for the implementation (if you like, the **code**). The sooner you get coding the better.

At University, there are lots of people can help design and build your future: peers, friends, academic staff, your careers service, employers and your wider social and professional networks but ultimately it is *your* responsibility to sort out whatever comes next. That might sound obvious but don't wait to get started later or for somebody else to do it for you, because it probably won't happen.

1.6 Your degree is not enough

You have worked hard to get the grades you needed to get into University. You've spent (or are spending) a significant amount of time and money studying your chosen discipline. You are really *geeking out* by going deep into your subject for a substantial period of time. Geekery, by which I mean being interested in a

**"there's only one person
who REALLY cares about
your career, and that's
you"**

Andy Stanford-Clark



Figure 1.3: There's only one person who REALLY cares about your career, and that's you. Quote via Andy Stanford-Clark (Stanford-Clark, 2019) to an unattributed IBM manager. Image of Andy by Gizmo~enwiki via Wikimedia Commons w.wiki/3TSn adapted using the Wikipedia app Thank you Andy for permission to use your photo.

subject for its own sake, is a *good* thing. Earning the title *geek* is a compliment, not an insult, and you should wear your geek badge with geek pride! Some people even say:

- The geeks will inherit the earth, see figure 1.4
- The (geeky) weirdness flows between us, anyone can tell to see us (Jr., 1988)

YOU are a geek, so where is *your* inheritance?

As a studious geek you might be tempted to believe that the world owes you something in return for all your geekiness. Unfortunately that's not the case.

At some point during or after your study, you might find yourself applying for a graduate job or graduate scheme. EVERYONE applying for these opportunities will have a degree or be rapidly on their way to getting one. So having a degree, even a really geeky one, isn't going to set you apart much from your competition. Even having a first class degree may not distinguish you that much from your competitors (Coughlan, 2019a; Borrett, 2019). Some employers would rather not know (or don't care) what University you went to, so your education might not make you stand out as much as you might like anyway. (Agnew, 2016; Garner, 2015)

What **WILL** distinguish you from your competitors is:

Geeks
Blessed are the ~~meek~~:
for they shall inherit the earth.

Matthew 5:5

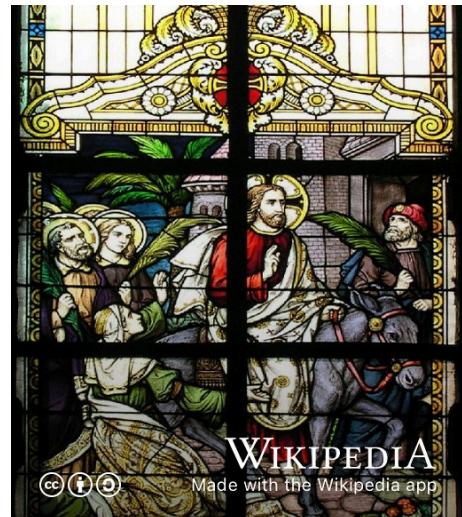


Figure 1.4: Blessed are the ~~meek~~ geeks, for they shall inherit the earth. (Anon, 1611; Robbins, 2012) You are a geek, so where is your inheritance? Image of stained glass window by Norbert Schnitzler via Wikimedia Commons w.wiki/43LN adapted using the Wikipedia app

- your **experience**: see chapter 7
- your **projects**: see section 6.6.4
- your communication skills:
 - Writing: see chapter 4
 - Speaking: see chapter 10
 - Listening: see chapter 20
 - Reading: see chapter 27
- your actions: what have you **done** with all of the above? How have you applied these fundamental communication skills to perform higher level communication tasks such as understanding, negotiating, persuading and leading etc. See examples of actions in chapter 18
- your **results** and **evidence**: see C.A.R.E in section 6.7.2: context, action, results and evidence. We're talking about all your results, not just exam results.
- any **leadership** you can demonstrate or **awards** that you have picked up along the way, see section 6.6.5

If you think that your degree will be enough to get you the job you want, bear in mind that:

1. There are more and more graduates, the UK for example recently passed the milestone of 50% of young people going into higher education. This compares to just 15% of over 18s who stayed in higher education in 1980 (Coughlan, 2019b)

2. While a degree is a *necessary* condition for joining a graduate scheme or taking a graduate job it is not a *sufficient* condition. Having a degree will not set you apart much from your competition, every applicant will have a Bachelors degree.
3. There are lots of graduates in your discipline. In the UK, for example, around 9,000 students graduate every year in Computer Science. If you're studying in the UK, what makes you different from the other 8,999 computer scientists graduating in your year?

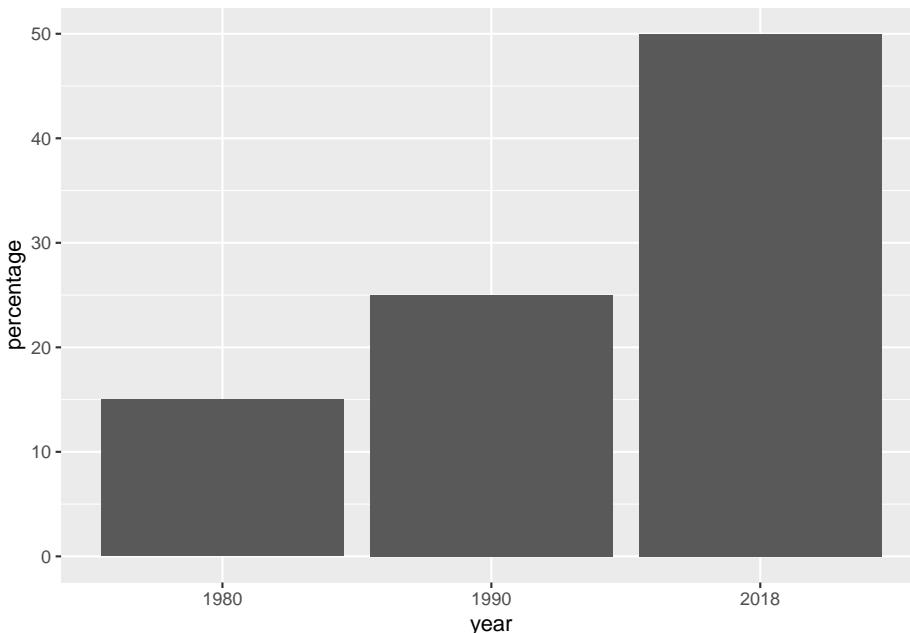


Figure 1.5: Percentage of young people in the UK going into higher education between 1980 and 2018. Over the last forty years, the proportion of young people going into higher education has more than doubled from 15% in 1980 to over 50% in 2018. Data taken from BBC news article on the symbolic target of 50% at university reached (Coughlan, 2019b)

As Sally Fincher puts it:

Computing is one of the largest subject areas in UK higher education, and is taught in almost every institution, graduating around 9,000 students every year — Sally Fincher (Fincher and Finlay, 2016)

Now, don't be disillusioned by the statistics because any degree can open doors to many careers in computing. Studying *computing* opens up plenty of doors, see figure 5.7. According to Charlie Ball, it is a myth that there aren't enough graduate jobs; one of four myths in the UK about the graduate labour market:

- **Myth 1:** *Everyone goes to university nowadays* : ~50% isn't everyone

(Ball, 2022a)

- **Myth 2:** *There aren't enough graduate jobs* (Ball, 2022a)
- **Myth 3:** *Some degrees have little value to employers*, see chapter 5
- **Myth 4:** *All the best graduate jobs are in London*, (or your local big city) see chapter 16

What the data in figure 1.5 show is that you'll need to look **beyond** your formal education to distinguish yourself from your competition. Your degree can certainly help you start a career, and computer geekery is a commercially valuable skill but a degree (however geeky) is typically not enough by itself.

There's plenty of graduate jobs for you to apply for, but that doesn't mean its going to be easy to walk into one when you graduate. Employers are looking for more from their employees than just *having* a degree, see chapter 7.

1.7 It's too late when you graduate

You might be tempted to postpone making difficult career decisions:

- I'll do it tomorrow ...
- I'll do it next week ...
- I'll do it next year ...
- I'll finish this assignment ...
- I'll finish this exam ...
- I'll finish this semester ...
- I'll finish my degree first, see figure 1.6 ...

Procrastination is a part of the human condition and its not necessarily a bad thing. Software engineer Paul Graham and journalist Oliver Burkeman describe three types of procrastination:

1. bad procrastination (Graham, 2005)
2. good procrastination (Graham, 2005)
3. better procrastination (Burkeman, 2021)

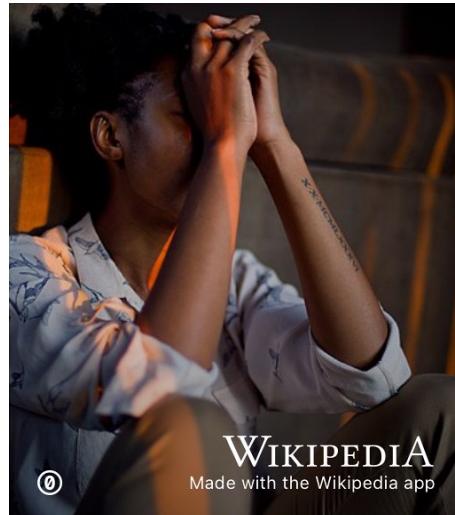
We all procrastinate because there's an infinite number of tasks we could be doing but a finite amount of time in which to do them.

Postponing decisions about your future is usually bad procrastination, because you risk neglecting the wrong things. You are relying on doing all your (career) testing at the end, when you graduate. You are procrastinating the **testing** stage of your career development, when its often better to test early and test often, so:

- “test-last (career) development” where you get experience at the end, *after* you graduate
- “test-driven (career) development” (TDD) where you get experience *before* you graduate

"I'll get my degree out of the way first then worry about jobs and careers when I finish University"

Procrastination



WIKIPEDIA

Made with the Wikipedia app

Figure 1.6: Procrastination: the attitude of “I’ll get my degree out of the way first then worry about jobs and careers when I finish University” is bad procrastination. It’s too late when you graduate to start thinking about what might come next (Graham, 2005) Stressed procrastinator picture by MismibaTinashe-Madando on Wikimedia Commons w.wiki/3TXo

It probably doesn’t help that many of the issues described and discussed in this book are typically not closely integrated into the curriculum in Higher Education. You’ll often find them on the edges, or completely outside of, standard University curricula.

Despite being sidelined, these issues matter and it is in your own self interests to start thinking about them right now. According to recent estimates by *Investors in People*, the average person spends **80,000 hours** working during their lifetime. (Investor, 2020) So, *whatever* you end up doing after University, you’ll be spending a lot of time doing it. Difficult decisions often get sidelined but it is never too early to *experiencing your future*, see chapter 7.

If you want to work for a big name like those in section 7.3.2 or 8.3.1, many of the larger graduate employers expect you to have *some* experience (see chapter 7) *before* you graduate. A large chunk of vacancies on graduate schemes are filled with people who have already been employed as interns or placement students within that (or another) organisation. So the sooner you start investigating employers by getting some experience the better decisions you’ll be able to make about what comes next. It’s (usually) too late when you graduate.

That doesn’t mean you have to know EXACTLY what you want to do when you finish. Lots of students don’t and I certainly didn’t when I graduated. I’d done a gap year teaching in India, two summer internships (in Sweden and the

United States) and a year-in-industry in the UK and I *still* graduated with **no clue** as to what I wanted to do next! The important thing is that you make a start by getting some experience, paid or otherwise, see chapter 7. Sometimes knowing what you **don't** want to do is just as valuable as knowing what you **DO** want to do. Either way, your future starts now, see figure 1.7.



Figure 1.7: Your Future starts now, not at some point later in time like when you graduate. Whatever stage of your study you are at, there's plenty you can do to start experiencing your future today, see chapter 7. Experience will make it easier to navigate your future. *THE FUTURE IS NOW* by Everydayphrase linktr.ee/everydayphrase

Computer scientists call this problem “search space reduction” (Ferrari et al., 2008). You have a feasible region of future possibilities and you need to narrow down the candidates. You could think of coding your future as an optimisation problem. Start optimising now, by getting some experience because it's too late when you graduate.

1.8 Yes, this WILL be on the exam

Students love to ask their teachers “*will this be on the exam*”? The short answer is **YES** (and **NO**)! Yes, this will be on the exam, but NO the exam won't be set by your University. Unlike other courses you've done, the examinations for this course aren't set by your University but by employers. Roughly speaking, there are three kinds of examinations that you'll need to get good at, shown in Table 1.1

Table 1.1: Examining your future: The “exams” used by employers, what gets assessed and the grades you can get. For written “exams” see chapters 4 and 6, for speaking “exams” see chapter 10 and for your employee “exams” see chapter 11.

Examination	What examiners are assessing	Grade
CV, application form covering letter	<ul style="list-style-type: none"> • Should we invite you to interview ? • Can you communicate well in writing? • Do you have any experience? • What interesting projects have you done? • See <i>debugging your future</i>, chapter 6 	pass/fail
Interview	<ul style="list-style-type: none"> • Should we offer you a job? • Would we want you on our team? • Can you communicate well verbally? • Can you communicate well nonverbally? • See <i>speaking your future</i>, chapter 10 	pass/fail
Employee performance	<ul style="list-style-type: none"> • Should we promote you? • Should we give you a pay rise? • Should we extend your contract? • See <i>surviving your future</i>, chapter 11 	pass/fail

So, *yes*, this will be on the exam, but *no*, the exams are obviously not set, administered, invigilated and marked by academics at your University. The exams are set by employers and the results are **brutally binary**:

- **PASS:** you’ve got the interview, job or promotion or...
- **FAIL:** none of the above. Next!

One of the challenging things about employers exams are, they typically do not have the bandwidth to give applicants useful feedback, other than a simple pass or fail. When it comes to job applications software engineer Gayle Laakmann McDowell calls this the “black hole”. The gravitational force of employers black holes is so strong that no CV or Résumé can escape, we’ll say more about this in chapter 6 on debugging your future.

It’s a similar story with interviews, if you fluffed an interview question or came across badly, it can be really difficult to find out from the employer what you did wrong. That makes it harder to learn from your mistakes.

GIMME SOME CREDIT

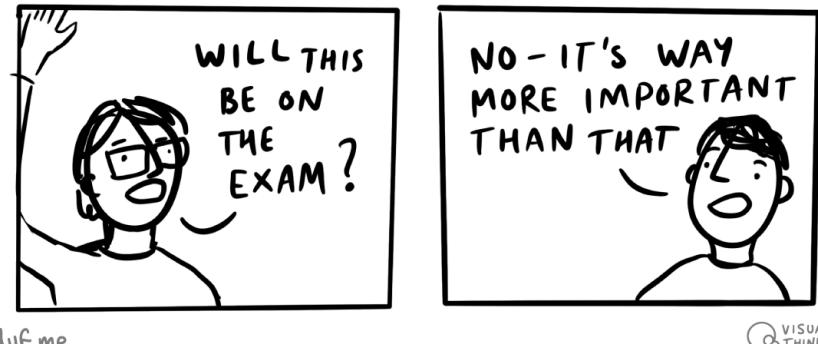


Figure 1.8: So *no* this will not be on the exam set by University, but *yes* it will be on the exams set by employers. Some of the most important exams you sit at (and after) University are set by employers. This guidebook will help you prepare for those exams and increase your chances of passing them. Gimme some credit figure by Visual Thinkery is licensed under CC-BY-ND

1.9 Practicing your future

There are practical exercises, for you to get your hands dirty with. Each chapter incorporates activities including individual exercises, group exercises, quizzes and points for wider discussion. You'll get a lot more out of this guidebook by doing the activities, rather than just reading it.

1.10 Navigating your future

There are **lots** of resources out there that offer self-help, career advice and techniques for self-improvement. It can be hard to know where to start, or even how to find your way around the mountains of advice.

Lots of professional advice is readily available, but how will you navigate it? This book provides signposts to what I think are the most important resources, each chapter has a signposts section, and they are all gathered together in the signpost at the end alongside everything (yes, **EVERYTHING!**) that this guidebook cites in the references, chapter 27.

1.11 Crediting your future

Get credit for your contributions. As well as being openly accessible on the web, this book is open source too. What this means is, you can contribute in



Figure 1.9: There are tonnes of resources out there offering advice on a huge range of professional issues. You can't read them all, but this guide aims to help you navigate the resources that will be most use to you

several ways described in section 0.8. All the written content for this guidebook is licensed under CC-BY-NC-ND, see the license in section 0.11.1.

1.12 Your future is different

I'm writing this guidebook because I need a resource for students to help them design, build, test and code their futures in computing. I need a book to help me teach to students to compete for jobs while at University, or shortly after graduating. I can't find anything suitable that meets all the requirements of the students I teach. Some problems with the resources that are already out there, and how this book addresses them, are shown in table 1.2

Table 1.2: *Coding your Future* is a guidebook that is different to all the other guidebooks

Other guidebooks	<i>Coding your Future</i> guidebook
<ul style="list-style-type: none"> • Expensive paper-based books • Inadequate linking, inbound & outbound • Poor citations • Often can't be searched 	<ul style="list-style-type: none"> • Free digital versions • Every link is clickable, every section, table and figure can be linked to • Everything cited, see section 1.12.1 • Search by any words you choose

Other guidebooks	<i>Coding your Future</i> guidebook
<ul style="list-style-type: none"> • Born printed: digital comes second • Websites are crippled “companions” to book • Ebook a clumsy afterthought • Video and audio is separate to book • Divorced from any curriculum • Not used for teaching, not embedded in curriculum • One-size-fits-all, generic, career and discipline agnostic • Institution centred: Institutionalised by Universities, employers or advertisers • Often excludes the student voice • Top down, corporation driven, formal, bland and sometimes impersonal • Closed source • Small number of authors & editors • Released late, released seldomly 	<ul style="list-style-type: none"> • Born digital: paper comes second • The website www.cdyf.me is the book • Ebook published simultaneously, see section 0.7 • Video and audio is integrated & embedded • Married¹ to the computing curriculum • Course textbook for COMP2CARS, embedded in the curriculum • Tailored to computing careers, see chapter 5, 9 and 20 • Student centred: agnostic to institutions, employers and advertisers • Includes student voice, see section 0.8 and chapter 20 • Bottom up, grassroots driven, informal, colourful and personal • Open source, see section 0.8 • Anyone can contribute or comment • Released early, released often, see section 1.12.3

To following sections give a bit more detail on the differences outlined in table 1.2.

1.12.1 Your future is signposted

Some career resources claim (or imply) that they are the *all you will need* to solve a particular problem or worse: solve *all of your problems!* Just buy this book, do this course, watch this video, listen to this podcast and all your problems will go away! Rather than continue this trend, this book **signposts** some of the most useful resources, see figure 1.10.

Scientists call signposting **citation**, so I've signposted and cited sources in this guidebook so that you can :

1. Follow them if the destinations are interesting or useful
2. Check and verify any facts and claims I make in this book for yourself

¹computationally wedded to the curriculum with kids and a mortgage

Signposts allow users to navigate on long journeys, sometimes over difficult terrain

Traffic sign



Figure 1.10: Wondering which way to go at the traffic sign? I've signposted the resources that will help you navigate the start of your professional journey. Which route will you take? Picture of a signpost in the Åland Islands, Finland by Sal via Wikimedia Commons w.wiki/3Xop adapted using the Wikipedia app

While this guidebook cites lots of resources, some of them are more important than others. Each chapter summarises these in a signposts section. You'll find everything else in the references, chapter 27. University and public libraries may also have physical and electronic copies of some of the books listed here.

I'm not suggesting that you read *all* these books right now, but that if a particular chapter has piqued your interest, these signposts are good places to keep going, if you haven't already read them. I hope you'll find these signposts handy for navigating the mountains of advice. Not all who wander are lost.

1.12.2 Your future is guided

This guidebook to your future accompanies a course that has been co-designed by students for students, with input from academics and employers. It unites several disparate themes into one coherent story, from fundamental questions about identity and wellbeing through to more applied and practical advice on job hunting, career progression and life after University. Resources that do this are typically scattered around in many different places. There is usually no narrative to tie them all together to help students navigate the mountains of advice as they embark on the first stages of their careers. I'll guide you through the narrative, but it is a *descriptive* story rather than a *prescriptive* one, see figure 1.11.

Although this is a course guidebook used in undergraduate teaching, you don't



Figure 1.11: Many formal education courses follow a *prescriptive pathway* (right), they lead you through a story from start to finish. This is the way of the University: “our courses” - our prospectus. This course is different, it follows a *descriptive pathway* (left) which means you pick your own way through the story. I’ll guide you through the story but you’ll need to choose *your way*: “my way”. You stood tall and faced it all. You did it your way. (Sinatra, 1969) Prescriptive vs. descriptive pathways sketch by Visual Thinkery is licensed under CC-BY-ND

need to be enrolled on the course to benefit from reading it, watching the videos and doing the exercises and coding challenges.

1.12.3 Your future is constantly updated

You are reading the alpha version, the Minimum Viable Product (MVP) of this guidebook, last updated on 03 December, 2022. That's software engineer talk for saying *it isn't finished yet*. Subsequent versions, will be continuously and iteratively released on a daily and weekly basis. They will include:

- More quizzes for better interactivity
- More videos on the Coding your Future YouTube channel
- Audio interviews with Students in the *Coding your Future podcast* in chapter 20
- More illustrations throughout the book
- Improved content, finish incomplete chapters
- Fix bugs and typos
- Your suggestions for improvements and corrections, via github etc see section 0.8

I'm taking a release early, release often (Raymond, 1999) approach to publishing this guidebook, you could call it agile book development or continuous book delivery, see figure 1.12 (Schuh, 2004)

1.12.4 Your future is personal

A lot of scientific and technical writing is written in the third person or passive voice, which is fine for academic writing, but can alienate readers. I have opted to use first person narrative where possible as it is shorter, and hopefully more engaging for you to read. (Poundstone, 2013) Where relevant, I've told personal stories to illustrate key points.

1.12.5 Your future has no paywall

You don't need to pay anything to read this book online because its not hiding behind a paywall, see section 0.11.1. Publishing this guidebook online makes it findable and accessible, something that isn't true of knowledge locked up inside other books.

Because this guidebook is online, it is searchable, browsable and linkable. You can link to whatever level you like, top level, chapter level and to every section and subsection level. Everything important has a Uniform Resource Locator (URL).

1.12.6 Your future has no login

You don't need to login to anything to use this guidebook either. Hurrah! One less password to remember.

"Agile: make it up as you go along. Waterfall: make it up before you start, live with the consequences"

Agile software development



Figure 1.12: Agile software developers make it up as they go along, whereas waterfalling software developers make it all up at the beginning and then live with the consequences. Quote via Paul Downey, thanks psd. It's the same with natural language engineering (books). I'm making it up as I go along, using agile book development methods. Women who code image by Justice Okai Allotey via Wikimedia Commons w.wiki/3Xnk adapted using the Wikipedia app

1.12.7 Your future is audible and watchable

This book is not just words and pictures, but includes audio and video as well, especially:

1. videos produced by third parties that are worth watching
2. audio produced by third parties that are worth listening to, either individual episodes or whole series
3. short videos produced by me, which augment the written content of this book, see the Coding your Future YouTube channel
4. the *coding your future* podcast which interviews undergraduate students, see chapter 20

1.13 Engaging with your future

I've tried to make the content of this book as engaging as possible by including pictures and conversations. *Your future* is deliberately playful and light-hearted. If you think this guidebook can be improved, let me know via the mechanisms described in section 0.8. I always welcome constructive feedback, especially when it comes via a pull request. Engage, see figure 1.13.

His catchphrases are
"Make it so" and
"Engage"

Jean-Luc Picard

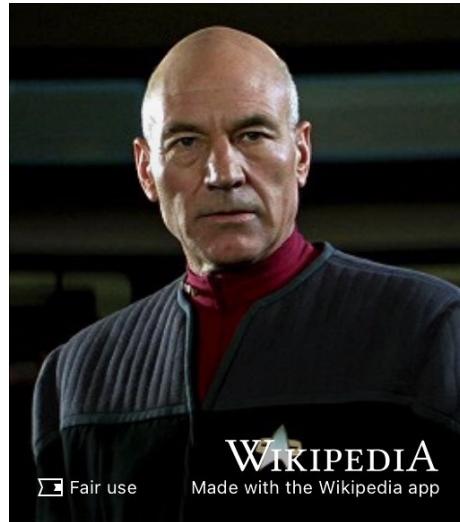


Figure 1.13: This is Captain Jean-Luc Picard of the Starship *Enterprise*. Engage! Fair use image of actor Patrick Stewart performing in *Star Trek* adapted using the Wikipedia app. Make it so.

1.14 Signposting your future

Each chapter in this book has a signposts section, highlighting key reading, watching or listening you could do next. This chapter has addressed the question of **why should you bother coding your future?** The answer is that your future is your responsibility and no-one else's. There are lots of people who can help shape your future, but none more than yourself. Software engineer Robert C. Martin argues this point in his book *The Clean Coder: A Code of Conduct for Professional Programmers*. (Martin, 2011)

What's good about *The Clean Coder* is that it is short (only 200 pages), well written and to the point. The main part of the book covers professional issues in software engineering, some of which I discuss further in chapter 11, *surviving your future*.

1.15 Summarising your future

If all that was too long, didn't read (TL;DR) for you, then you'll be relieved to hear that each chapter has a TL;DR (executive) summary.

TL;DR an abbreviation
for "Too Long; Didn't
Read"

Reading

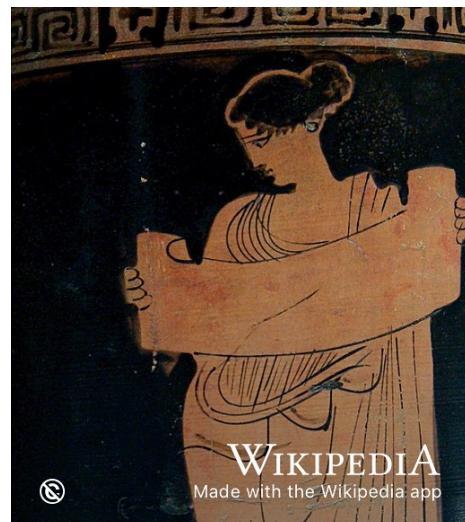


Figure 1.14: There's always more stuff you should be reading. If this guidebook is a bit **Too Long, Didn't Read** (TL;DR) then each chapter has a brief summary at the end. Public domain picture of ancient greek muse reading a scroll (that's probably too long) via Wikimedia Commons w.wiki/3Xoh adapted using the Wikipedia app

The TL;DR for this chapter is, you should read this guidebook because it is different to all the other guidebooks. It will help you take responsibility for maximising your future. No-one else is going to do this for you. Your degree

will help open up future options, but it is not enough by itself so you'll need to maximise the return on your investment. Don't procrastinate because it's too late when you graduate and *YES* this will be on the exam (set by future employers). This guidebook has signposts to help you navigate, design, build, test, debug and code your future in computing.

It looks like the reboot has finished, so we're ready to go. In the next chapter, you will reflect on who you are. What's your story, coding glory?

Chapter 2

Knowing your future

Hello, who are you? What's your story, coding glory? What are you good at, what do you like doing and what do you value? What are your hopes and dreams for the future? Tell me about your **education** and who you are. What unique talents are you finding and developing during your education? How are you striving to become a better version of you? Having good self knowledge will help you answer these big questions, which are important for your future. Knowing your future depends on knowing who you are now.

2.1 What you will learn

Reading this chapter and doing the activities will help you to

1. Develop better self-awareness by experimenting with seven techniques for exploring different dimensions of your identity:
 - Protected characteristics
 - What's your story, coding glory?
 - Head, heart and hands
 - Ikigai: reason for being
 - Personality profiling
 - Privilege audit
 - Deathbed thought experiment
2. Know yourself better so that you can:
 - start coding your future
 - articulate who you are to potential employers

2.2 Improving self-awareness

Knowing who you are now, will help you explore and understand who you might become in the future. That future you is the best possible a *better* version of

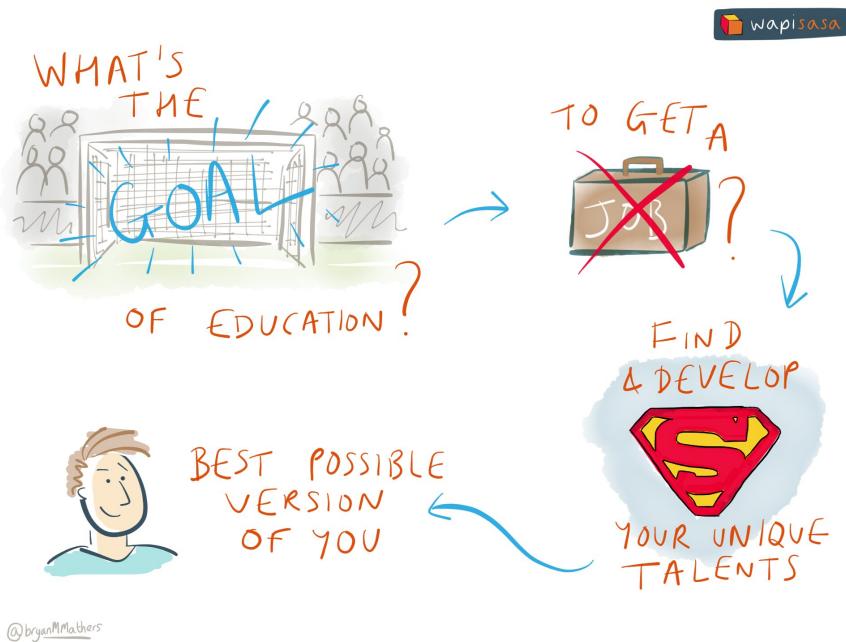


Figure 2.1: Your education is a crucial part of your story and who you are. The purpose of your education is not *just* to get you a job but to also find and develop your unique talents. What are your unique talents? How are you developing them as part of your broader education to become the best possible a *better* version of yourself? Goal of Education sketch by Visual Thinkery is licensed under CC-BY-ND

you shown in 2.1. Since you started it as a small child, your education has been:

- about bettering yourself
- a key part of your identity
- inherently communal and social

At University only a fraction of your education will happen during formal lectures, labs, tutorials and seminars see figure 2.2

**"Education is the sum of
what students teach
each other in between
lectures and seminars"**

Stephen Fry



WIKIPEDIA
Made with the Wikipedia app

Figure 2.2: According to Stephen Fry, Education is the sum of what students teach each other in between lectures and seminars. (Fry, 2010) So you can get more out of your short time at University by actively engaging with your peers with peer learning, peer instruction, peer support and informal learning. You, and your fellow students, will both benefit by teaching and learning from each other. Public domain portrait of Fry at Winfield House in 2016 by the US Embassy in London w.wiki/4wrn

Your identity, who you are, is complex, dynamic and high-dimensional so you will need several different techniques to improve your self-awareness. Here are seven techniques to get you started:

1. your protected characteristics, see section 2.2.1
2. your story so far, see section 2.2.2
3. your head, heart and hands, see section 2.2.3
4. your reason for being (*ikigai*), see section 2.2.4
5. your personality profile, see section 2.2.5
6. your privileges, or lack of them, see section 2.2.6
7. your deathbed *thought experiment*, see section 2.2.7

2.2.1 Your protected characteristics

Some of your identity includes characteristics that are protected. In the UK, the Equality Act of 2010¹ protects you from discrimination at work or in education, based on what are known as “protected characteristics”. (Servant, 2020a). This means that:

- Your **age** should not determine how you are treated
- Your **disabilities** should not determine how you are treated
- Your **gender** should not determine how you are treated (Saini, 2018; Damore, 2017; Lewis, 2017; Bates, 2016)
- Your **gender re-assignment** should not determine how you are treated
- Your **marriage** or civil partnership should not determine how you are treated
- Your **pregnancy** and maternity should not determine how you are treated
- Your **race** (including colour, nationality, ethnic or national origin) should not determine how you are treated (Eddo-Lodge, 2017; Saini, 2019)
- Your **religion** or beliefs should not determine how you are treated
- Your **sex** should not determine how you are treated (Price, 2019)
- Your **sexual orientation** should not determine how you are treated (Britton, 2019)

These are a special part of your identity because they are protected. This means they shouldn't be a factor in apply for a job or a subject for discussion in a job interview. Let's look at gender, sex and sexual orientation in more detail because they can be easily misunderstood. One way to understand gender better is to break it down into four characteristics:

1. **Identity**: your gender identity
2. **Expression**: your gender expression
3. **Sex**: your biological sex
4. **Attraction**: your sexual and romantic orientation

These protected characteristics are often conflated, because people tend to confuse gender, sex and sexual orientation. They are *not* the same thing as shown in equation (2.1)

$$\text{Identity} \neq \text{Expression} \neq \text{Sex} \neq \text{Attraction} \quad (2.1)$$

Lets have a look at each of those in turn:

1. **Identity** is who you know yourself to be in your own head. Gender identity is based on how much you align (or don't align) with the options for gender based on your psychological sense of self. This includes, but is not limited to:
 - Woman | Genderqueer | Man see en.wikipedia.org/wiki/Gender_binary and en.wikipedia.org/wiki/Non-binary_gender

¹<http://www.legislation.gov.uk/ukpga/2010/15/contents>

2. **Expression** is how you demonstrate your gender based on gender roles through the ways that you act, dress, behave and interact. These are not exclusive categories, you might choose to express yourself in all three ways including:
 - Feminine | Androgynous | Masculine see en.wikipedia.org/wiki/Gender_expression
3. **Sex** is often conflated with gender. Sometimes called anatomical sex or physical sex, your biological sex is objectively measurable using features such as your genitalia, chromosomes (your sex-determination system XX, XY etc), hormones, body hair, ovaries and testes. Your biological sex includes, but is not limited to:
 - Female | Intersex | Male see en.wikipedia.org/wiki/Sex
4. **Attraction** is who you are physically, spiritually and emotionally attracted to. Like biological sex, sexual orientation is often conflated with gender but isn't always a component of gender. We categorise the attraction we experience in gendered ways. Your sexual orientation includes, but is not limited to:
 - Asexual | Heterosexual | Bisexual | Homosexual see en.wikipedia.org/wiki/Sexual_orientation and [en.wikipedia.org/wiki/Romance_\(love\)](https://en.wikipedia.org/wiki/Romance_(love)) and en.wikipedia.org/wiki/Asexuality

These characteristics are summarised in the genderbread person shown in figure 2.3

Gender is probably more complicated than you realised, but the framework above will help you understand it better or help you explain the subtleties to someone else by breaking a complicated concept into bite-sized, digestible pieces.

2.2.2 What's your story, coding glory?

The protected characteristics described above are the part of your story you'll be most aware of. But there's much more to your story than these characteristics. We're hardwired to love storytelling because it helps us understand the world around us, see figure 2.4. We use stories to organise and communicate, so knowing your story is a crucial part of knowing who you are and articulating that to other people.

So, *what's your story, coding glory?* (Gallagher, 1995)

Self-awareness, understanding who you are, is important for leading a healthy and happy life, and likely to be an important factor in your future success. One way to develop better self-awareness is to think about the finer details of your story. (Box and Mocine-McQueen, 2019)

- How did you get here?
- What has inspired you?
- Where are you going next?
- Who is the authentic you? (Ware, 2011)
- What are your hopes and dreams?

By starting to answer these (admittedly **BIG!**) questions you will gain a better

The Genderbread Person^{v4} by itspronouncedMETROsexual.com

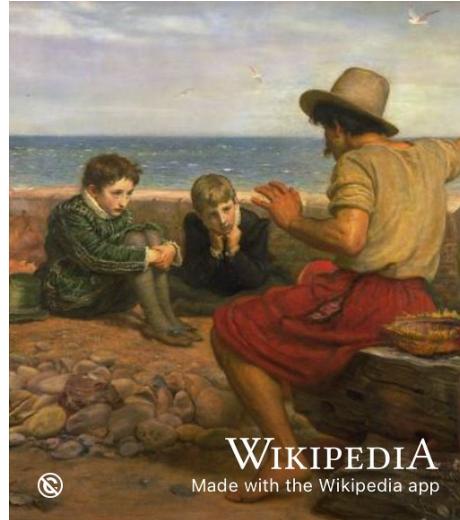


Genderbread Person Version 4 created and uncopyrighted 2017 by Sam Killermann [For a bigger bite, read more at www.genderbread.org](http://www.genderbread.org)

Figure 2.3: Decomposing gender into its components of identity (woman-ness & man-ness), attraction (romantically and sexually), expression (femininity and masculinity) and sex (female-ness & male-ness). Genderbread Person v4.0 (uncopyright) by Sam Killerman at genderbread.org and itspronouncedmetrosexual.com

A need to tell and hear
stories is essential to the
species *Homo sapiens*

Storytelling



WIKIPEDIA

Made with the Wikipedia app

Figure 2.4: Storytelling is an ancient art and who doesn't love a good story? As a species *Homo sapiens*, we need to tell and hear stories to understand the world around us. What's your story, coding glory? Public domain image of a painting by John Everett Millais, with a seafarer telling the story of what happened out at sea, via Wikimedia Commons [w.wiki/3VHM](https://commons.wikimedia.org/w/index.php?curid=3VHM)

understanding of who you are. This includes your strengths, weaknesses, motivation and values. (Bolles, 2019) Your story is complex but you need to know it so you can distil the details into *much* shorter stories on your job applications described in section 6.6. Crucial parts of your story are:

- **Characters:** *Who* are the key people who have influenced your story so far?
- **Settings:** *Where* and *when* have your stories taken place?
- **Conflict and change:** Education means bettering yourself which inevitably involves change so:
 - *How* have your experiences changed you?
 - *What* did you learn from these experiences?
- **Emotions:** how did you *feel* at the time? Fearful, happy, excited, surprised, empowered? Looking back, how do you feel about these stories now? Have your feelings changed and if so, why?

Write these stories down. By reflecting on them and articulating them, you will improve your self awareness. Ultimately, some of these stories can eventually be distilled down into content on your CV (with emotions edited out!), see section 6.7.2.

2.2.3 Head, heart, hands

Another technique for building your self-awareness is to reflect on your knowledge, values and skills. In Waldorf education and Montessori education this is characterised as “head, heart and hands” outlined below and in figure 2.5 (Easton, 1997)

1. **Head:** What do you *know*?
2. **Heart:** What do you *value*, what motivates you?
3. **Hands:** What can you *do*? What have you *done* so far? What will you do with your skills in the future? Your actions define your impact, see chapter 18

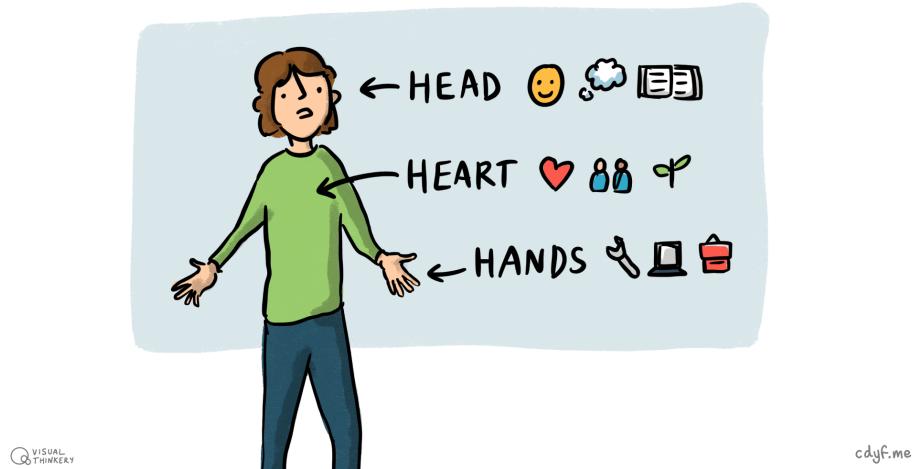


Figure 2.5: What knowledge do you have (**head**)? What are your values (**heart**)? What skills and experience do you have (**hands**)? Head, heart, hands sketch by Visual Thinkery is licensed under CC-BY-ND

Again, answering these questions will help you understand and articulate your story.

2.2.4 Ikigai: Reason for Being

Many of the learning outcomes described above are non-trivial. You may have good self-awareness and be able to describe aspects of who you are in a matter of minutes. Other personality traits make take longer to uncover. You can develop better self-awareness by describing four attributes shown in Figure 2.6, together these are known as your ikigai (いきがい) or “reason for being”.

- what do you love doing?
- what are you good at?

- what does the world need?
- what can you be paid for?

You'll be lucky if you can find activities at the intersection of all four sets (*ikigai*) shown in Figure 2.6. In practice, you may realistically only be able to achieve one, two (the **Passion, Mission, Profession and Vocation**) or three intersections. That said, it's still a valuable exercise to think about what is in each set for you.

Take a sheet of paper, draw the four overlapping rings shown in Figure 2.6, and spend five to ten minutes adding things in each ring.

- What are your values?
- What motivates you?
- Are there things you like doing that you aren't particularly good at?
- Why does that make them enjoyable?

Thinking about your *ikigai* will clarify your knowledge of yourself.

2.2.5 Personality profiling

Personality tests are designed to reveal your key characteristics, see figure 2.8. Besides helping you develop better self-awareness, they are used by some employers during the interview. Personality tests are often part of a wider set of aptitude and psychometric tests to test measure your ability at numerical reasoning, verbal reasoning, logical reasoning and situational judgement.

If you haven't done a personality test before, the 16 short questions at icould.com/buzz-quiz will give you a flavour of what to expect. Which animal are you? There are lots of tools for personality profiling which go into more depth by asking you a lot more than 16 questions. Some of these services are free such as:

- 16personalities.com: similar to Myers–Briggs Type Indicators (MBTI)
- strengthsprofile.com
- practiceaptitudetests.com/personality-test
- jobtestprep.co.uk/free-personality-test

Different personality tests try to measure various dimensions of your personality such as:

- your relationship to other people
- your emotional intelligence
- your organisational skills
- your openness to new experiences
- your resilience and mindset

Your University may also pay for subscription-based personality profiling services, check with your careers service for details.

Q VISUAL
THINKERY

cdyf.me

Figure 2.6: Reasons for being, a concept in Japanese known as *ikigai*. According to *ikigai*, the most meaningful life lies at the intersection of four sets: (1) What you are good at, (2) What you love, (3) What the world needs and (4) What you can get paid for. What do *you* have in each of these sets and what are on your personal intersections? *Ikigai* sketch by Visual Thinkery is licensed under CC-BY-ND



Figure 2.7: How well do you know yourself? Know who you are sketch by Visual Thinkery is licensed under CC-BY-ND

A personality test is a method of assessing human personality constructs

Personality test

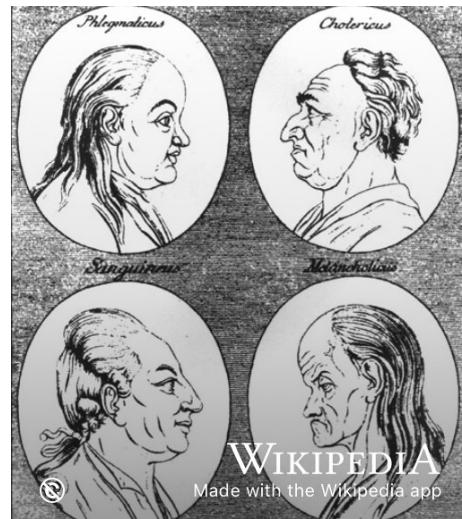


Figure 2.8: Personality profiling has been around for a long time. Early theories proposed there were four temperaments: 1. phlegmatic 2. choleric 3. sanguine and 4. melancholic shown in this 18th century illustration. Public domain image of a woodcut by Johann Kaspar Lavater via Wikimedia Commons w.wiki/5xmG adapted using the Wikipedia App

2.2.6 Privilege audit

If you have privileges, it is important to recognise and acknowledge any advantages these have given you in life. They are a key part of your identity, who you are and your ego. If you don't recognise your privileges then you don't know yourself. Ask yourself *honestly*, what privileges do you have? For example, is it *just* your skills and knowledge that have got you into higher education, or have you been fortunate?

- Don't have too high an opinion of yourself and your privileges, otherwise you'll overlook your flaws and won't improve
- Be wary of your ego, or as Chris Anderson puts it in his guide to public speaking: "park your ego" (Anderson, 2016)
- If you're struggling to think of any privileges, see section 2.4.4 for some suggestions.

Being mindful of any privileges that you have is not just a part of your identity. Being grateful for those privileges is beneficial for your mental health too, see the discussion of section 3.4.

2.2.7 Deathbed thought experiment

Imagine for a moment you are on your deathbed. Not at some point in the future, but right now. Your heart (see figure 2.9), which has served you well until now, starts behaving strangely. You start having palpitations, hearing your own heartbeat (Hütter and Hilpert, 2003) and worry what's going on with the muscle responsible for keeping you alive. Worst case scenario is, your time is up. This happened to me and its a useful thought experiment to force you to think about what matters in your life. (Hull, 2021b)

One of the things you'll probably want to do is reflect on your life and wonder:

- What did you achieve?
- Do you have any regrets, if so what are they?
- What would you change if you could carry on living?

This can be a useful technique for forcing you to think about who you are and what you value. If you find this activity difficult, see section 2.4.1 for some hints. The Roman's called this *Memento mori* which roughly translates from latin as "remember that you are mortal" or "remember that you have to die".

Co-founder of desana.io Michael Cockburn argues that you should "make decisions as if you were on your deathbed". (Cockburn, 2022) Remember that you have to die. *Memento mori*. Do you realise that everyone you know someday will die? Instead of saying all of your goodbyes, let them know you realise that life goes fast. It's hard to make the good things last. (Coyne et al., 2002)

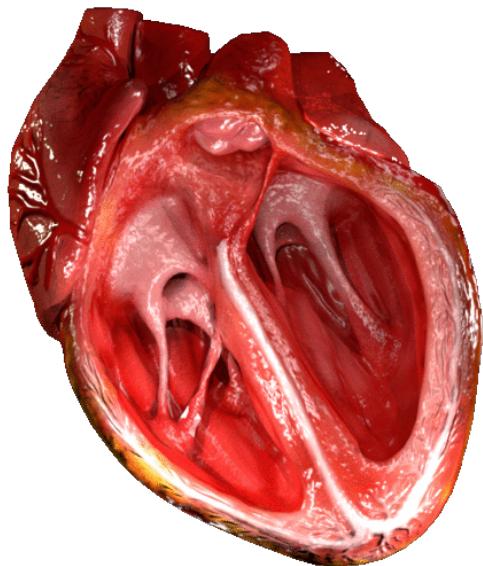


Figure 2.9: One day your heart will stop beating, hopefully not for a long time yet, but it will happen. If you live for an average length of time, your heart will beat more than 2.5 billion times. Now imagine your heart will stop beating in a weeks time. As you reflect on your life, would you have any regrets? If so, what would they be? This *deathbed thought experiment* will force you to think about who you are and what you value. Computer generated cross section 3D model of the human heart via Wikimedia Commons by DrJanaOfficial at [w.wiki/36zc](https://commons.wikimedia.org/w/index.php?title=File:Human_heart_cross_section_3D_model_by_DrJanaOfficial.jpg&oldid=3620000)

2.3 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

This chapter has looked at some big issues around identity, by inviting you to think about some fundamental questions. Another way to think about these questions is as coding challenges. They are non-trivial questions to answer, it might take you weeks, months or even years to answer some of them. But they are worth spending time thinking about

- What are your values?
- What makes you happy?
- What do you want to get from your time at University?
- What do you want after University?
- Where do you see yourself in x years time?
- What are your privileges (if any), see section 2.4.4

The signposts in the next section may help tackle some of these coding challenges.

* RESUME

2.4 Signposts from here on identity

This chapter challenges you to reflect on who you are and what you're good at. We've only scratched the surface, so if you want to dig deeper you'll find the following resources useful:

- *The Top Five Regrets of the Dying*
- *What Colour is Your Parachute?*
- *How Your Story Sets You Free*
- A range of books about privilege

2.4.1 Your dying regrets?

One of *The Top Five Regrets of the Dying* (Ware, 2011) is that people wish they'd had the courage to live a life true to themselves, and not a life that others expected of them. Figuring out exactly who your authentic self is can be challenging. Bronnie Ware's book might help, it has some very moving, personal and insightful true stories of the regrets that people have that will illuminate your own values. The top five regrets, outlined in the book are:

1. I wish I'd had the courage to live a life true to myself, not the life others expected of me
2. I wish I hadn't worked so hard
3. I wish I'd had the courage to express my feelings

4. I wish I had stayed in touch with my friends
5. I wish that I had let myself be happier

You need to be courageous to live a regret-free life but the alternative is to die full of regret, see Bronnie's video in figure 2.10.

Figure 2.10: Palliative care nurse Bronnie Ware explains the top five regrets of the dying. Bronnie learned a lot from looking after people on their deathbeds, then wrote it all down in a fantastic book (Ware, 2011). You can also watch the two minute video embedded in this figure at youtu.be/nayz3xJxRTA (Ware, 2017)

2.4.2 Colouring your parachute

Since first being published in 1972, over ten million copies of *What Colour is Your Parachute?* have been sold. It has been translated into 20 languages and is used in 26 countries. What is good about *Parachute* is that it has some useful *self-inventory* exercises that go beyond the introductory ones in this guidebook, particularly in the context of your future career. While the style and examples can be U.S. centric, it's a classic self-help book that looks at a broad variety of issues around job hunting. The author, Richard Nelson Bolles was a Harvard educated chemical engineer and explains how to work out what you'll be doing in five years time in the video in figure 2.11. He looks at techniques for working our

your direction including random luck, intuition and improving self-awareness, using similar techniques to those described in this chapter.

Figure 2.11: Where will you be five years from now? Best-selling author of *Colouring Your Parachute* Dick Bolles discusses the gaps between education and employment. You can watch also watch the full 32 minute video embedded in this figure at youtu.be/oeP6Pm3Xf-8 (Bolles, 2014)

2.4.3 What's your story?

A useful technique for developing self-awareness is to think about what your story is. Heather Box and Julian Mocine-McQueen's book *How Your Story Sets You Free* (Box and Mocine-McQueen, 2019) takes a storytelling approach to help you gain a better picture of who you are and what you value. What's good about this book is its short, less than 100 pages and contains practical exercises which extend those in this chapter.

2.4.4 Check your privileges

Reflecting on your identity should lead you to check any privileges you might have. Being grateful for any privileges you may have is also beneficial for your mental health which we talk about in chapter 3 so:

- **If you are white** a good place to start understanding your white privileges is *Why I'm No Longer Talking to White People About Race* by Reni Eddo Lodge (Eddo-Lodge, 2017) and *Superior: The Return of Race Science* by Angela Saini
- **If you are male** a good place to start understanding the privileges you have as a result of being a man is *Inferior* by Angela Saini (Saini, 2018)
- **If you are socially privileged** a good place to start understanding the privileges you have as a result of your class is *The Class Ceiling: Why it Pays to be Privileged* by Sam Friedman and Daniel Laurison (Friedman and Laurison, 2020).
 - If you were privately (or selectively) educated in Britain (or elsewhere) you should read *Engines of Privilege: Britain's Private School Problem* (Green and Kynaston, 2019) or *Posh Boys* by Robert Verkaik (Verkaik, 2019)
- **If you are heterosexual** a good place to start understanding the privileges you have as a result of your sexual orientation is Ben Britton's presentation on *No sexuality please, we're scientists* (Britton, 2019) which covers bisexuality and homosexuality, including lesbian and gay homosexuality
- **If you are gender binary** a good place to start understanding the privileges you have as a result of being gender binary is Ben Britton's presentation (Britton, 2019) which also covers transgender, genderqueer, non-binary and plus identities.

If you think you got to where you are purely because of your talents, think again: there's good evidence to show that luck plays a much bigger role than many of us would like to imagine. (Pluchino et al., 2018) Luck is as much a part of your identity as your talents.

There is a lot more to your identity than your race, class, gender and sexual orientation, see your protected characteristics in section 2.2.1.

2.5 Summarising self awareness

Too long, didn't read (TL;DR)? Here's a summary:

Know thyself was one of three maxims inscribed on the Temple of Apollo in Delphi, in ancient Greece, see figure 2.12. It's still an important maxim thousands of years later. To know thy future, you must first know thyself.

This chapter has looked at knowing yourself and who you are. Being self aware, understanding your strengths and weaknesses is key to getting what you need from your future. Questions about your identity are non-trivial, hopefully this chapter has started you thinking about who you are, what motivates you and what you want out of life. You will need to keep revisiting these questions about your identity because some aspects of your identity may change over time.

the first of three Delphic maxims inscribed in the forecourt of the Temple of Apollo at Delphi

Know thyself

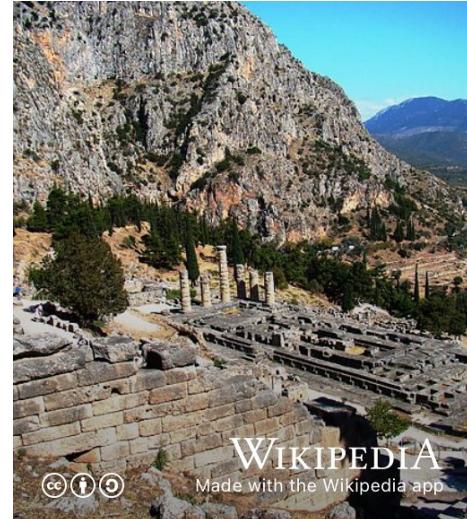


Figure 2.12: *Know thyself* was an important maxim in ancient Greece, written on temples like this one in Delphi. How well do you actually know yourself? Picture of the ruins of the Temple of Apollo at Delphi by Edward Knapczyk on Wikimedia Commons w.wiki/5MEd adapted using the Wikipedia App

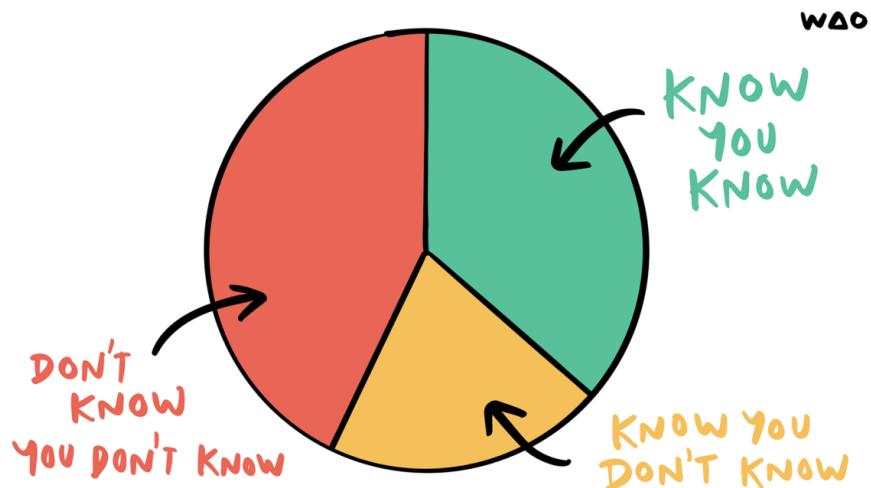


Figure 2.13: You know what you know about your identity and who you are. What parts of your identity do you know you *don't* know? Are there things about you that you *don't know* that you *don't know*? What are the unknown unknowns about you? What you don't know you don't know sketch by Visual Thinkery is licensed under CC-BY-ND

What do you know and what don't you know about yourself, see figure 2.13? These fundamental design questions you'll need to address when you starting building your future. This chapter has looked at some simple techniques for exploring your unknown unknowns.

Knowing your future is Coding your Future.

In the next chapter, we'll look at nurturing your future with a focus on mental health.

Chapter 3

Nurturing your future

It doesn't matter if you are a student, an employee or even both at the same time. To be successful at studying or working, you need to take your well-being seriously. By well-being, I mean your health and happiness. Your health isn't just about your physical health but also your mental health and the two are very closely linked. It's all too easy when you are busy or stressed to neglect your well-being and then **bad-stuff™** happens. This chapter looks at your well-being, and how you can nurture it. Because looking after yourself now will also nurture your future.

3.1 What you will learn

By the end of this chapter you will be able to:

1. Identify some of the symptoms of mental ill health in yourself and your peers, particularly anxiety and depression
2. Describe five self-help techniques for improving mental health
3. Describe services and other people you can approach if you (or someone you know) is being affected by mental ill health and self-help isn't enough
4. Schedule activities for improving mental and physical health into your daily or weekly routine

3.2 Stressed, depressed and exam obsessed?

One of the main target audiences for this textbook is Generation Z, that's people born sometime between the mid-nineties to early 2000's. There's some evidence to suggest that Gen-Z-ers like you have a higher probability than previous generations of being, as *The Economist* puts it:

- Stressed



Figure 3.1: Enduring Turing: You probably already knew that Alan Turing was an outstanding Computer Scientist, but did you know he was also a respectable athlete too? Turing ran, cycled and rowed to relieve stress, and came close to competing in the Olympics as a runner (Kottke, 2018). This should come as no surprise, the connections between well-being and academic performance are widely documented. Image via Jonathan Swinton's biography *Alan Turing's Manchester*. (Swinton, 2019) The copyright holder for this image has been unidentifiable or unresponsive at their self-advertised contact details.

- Depressed
- Exam obsessed

Are you stressed? Depressed? Exam obsessed? Are you more worried about getting good grades than drinking too much or having unplanned pregnancies? According to *The Economist* members of Generation Z are more likely to be affected by mental health issues. (Anon, 2019a) If that's you then you are not alone. Stress, depression and results obsession have been around a while. Many people struggle with them so let's look at them in a bit more detail.

Coding Caution

I am neither a medical doctor or a psychologist!

This chapter is based on my personal experiences of clinical depression, antidepressants and the experience of close friends and colleagues who have been affected by generalised anxiety disorder. So this chapter gives you a quick overview of mental health and points you to where you can find out more.

If you are affected by mental ill health, you should speak to a trained professional.

3.3 Mental ill health

Stress can lead to many kinds of ill health. Turing was put under lots of stress by his government bosses, people like Alastair Denniston and Stewart Menzies. (Tyldum, 2014) When asked why he punished himself so much in training (see figure 3.1), Alan Turing's reply is shown in figure 3.2.

University is a positive experience for many people, however like Alan, you may also experience periods of stress. This may also be accompanied by anxiety, loneliness and depression. Financial, social and academic pressures alongside concerns about employability, fallout from COVID-19, war in Ukraine and climate change etc can all have an impact on your wellbeing. Statistically, one in four of us will be affected by mental ill health during our lifetime. Two of the most common forms of mental ill health are:

- **Anxiety:** persistent feelings of unease, such as worry or fear
- **Depression:** a low mood that *lasts for a long time* and affects your everyday life. Being affected by depression is **not** the same as feeling sad

The *persistent* and *lasting a long time* are important here because while it's part of the human condition to worry and feel sad, that doesn't *necessarily* mean you are affected by poor mental health. How long is a long time? Definitions vary, but around two weeks is a good rule of thumb. There are many forces (F) at play, feelings of sadness or anxiousness are normally balanced out by counteracting feelings of calm or happiness shown in figure 3.3:

- Forces that make you unhappy: F_{SAD}

"I have such a stressful job that the only way I can get it out of my mind is by running hard; it's the only way I can get some release"

Alan Turing

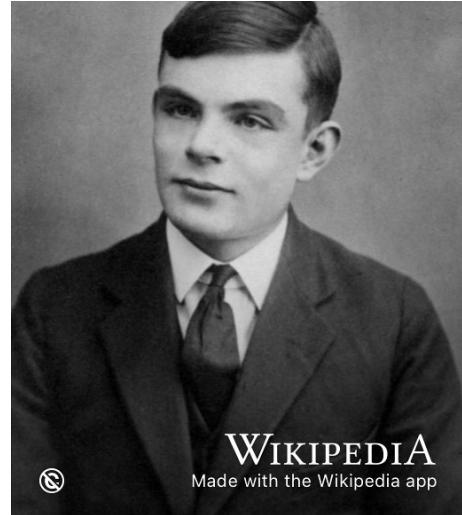


Figure 3.2: When asked why he trained so hard, Alan Turing replied: “I have such a stressful job that the only way I can get it out of my mind is by running hard; it’s the only way I can get some release”. Like many, Turing found running a relief from the mental pressures he was under in his job. (Kottke, 2018) Studying can be stressful too and put you under pressure. Your academic performance at University can be significantly improved by taking regular exercise and it will improve your mental health too. Public domain portrait of Alan Turing aged 16 via Wikimedia Commons w.wiki/oZx adapted using the Wikipedia app

- Forces that make you happy: F_{HAPPY}
- Forces that make you anxious: F_{ANXIOUS}
- Force that make you calm: F_{CALM}

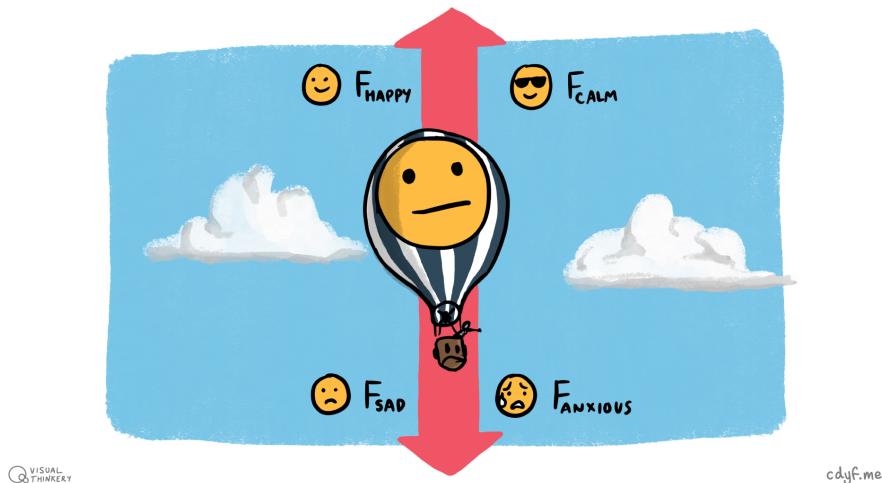


Figure 3.3: Your mental health is like a balloon, acted on by forces of anxiety, calmness, happiness and sadness. These opposing forces normally balance each other out over time. Crudely speaking: $(+) + (+) = .$ Good mental health means your balloon stays airborne, despite the inevitable up and downs. If you are affected by poor mental health, feelings of anxiety or sadness will outweigh their counterparts, leading to *persistent* anxiousness or depression. This will cause your balloon (mental health) to “crash”. This chapter introduces what these mental health issues are and what you can do if you, or someone you know, is affected by poor mental health. Free body diagram by Visual Thinkery is licensed under CC-BY-ND

It might help to consciously list what these forces are so that you are more mindful of them.

3.3.1 Anxiety

Anxiety is one of most common mental health disorders and can lead to depression, increased risk of suicide. Generalised Anxiety Disorder (GAD), a common form of anxiety is explained in the video in Figure 3.4 and at [nhs.uk/conditions/generalised-anxiety-disorder/](https://www.nhs.uk/conditions/generalised-anxiety-disorder/). People who are affected by anxiety may struggle to function normally, and find routine everyday task difficult or impossible.

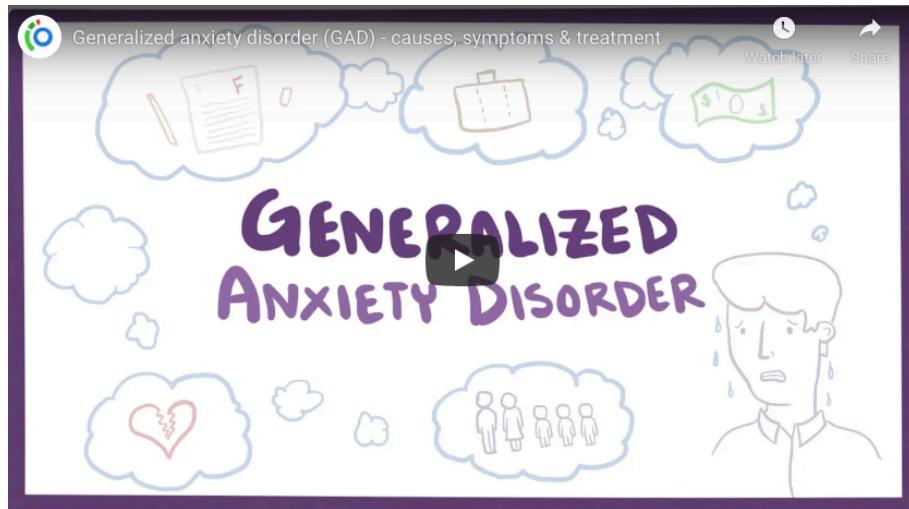


Figure 3.4: Generalised anxiety disorder is a condition characterised by excessive, persistent and unreasonable amounts of anxiety and worry about everyday things. Note that the video takes an American perspective using American terminology such as DSM-5. The image in this figure is a screenshot, you can watch the full video at youtu.be/9mPwQTiMSj8 (Desai, 2016)

3.3.2 Depression

Millions of people around the world live with depression. If you are affected by depression it can be really hard to talk about it as there are many social stigmas around mental health. Thankfully depression is largely preventable and treatable. Recognising depression and seeking help is the first and most critical step towards recovery. To mark World Mental Health Day writer and illustrator Matthew Johnstone tells the story of how he overcame the “black dog of depression” in the video in Figure 3.5 made in collaboration with the World Health Organization (WHO).

A common issue with poor mental health is that the people affected often don't realise they need help, partly because of the social stigmas that prevent people talking about mental health in the first place. Computer Scientist Dave Cliff explains how he recognised he was being affected by poor mental health in figure 3.6

Just *talking* about mental health issues is an important step in removing the social stigmas that surround them.

3.3.3 Drugs

Prescription medication can help some people with their mental health. For example, when I was affected by depression, Selective Serotonin Reuptake In-

Figure 3.5: Matthew Johnstone explains how he overcame the affects of depression, using the metaphor of the black dog. You can also watch the watch the 4 minute video embedded in this figure at youtu.be/XiCrniLQGYc (Johnstone, 2012)

Figure 3.6: “When you haven’t slept a wink for five days, the madness arrives in the post very quickly afterwards”. Dave Cliff at the University of Bristol explains how he deals with poor mental health, severe anxiety, depression and suicidal thoughts. (Cliff, 2018a) You can also watch the 7 minute video embedded in this figure at youtu.be/OtELvV8-OK0 (Cliff, 2018b)

hibitors (SSRIs) worked for me, shown in Figure 3.7, but they don't for everybody. Sometimes the drugs don't work, they just make you worse. (Ashcroft, 1997)

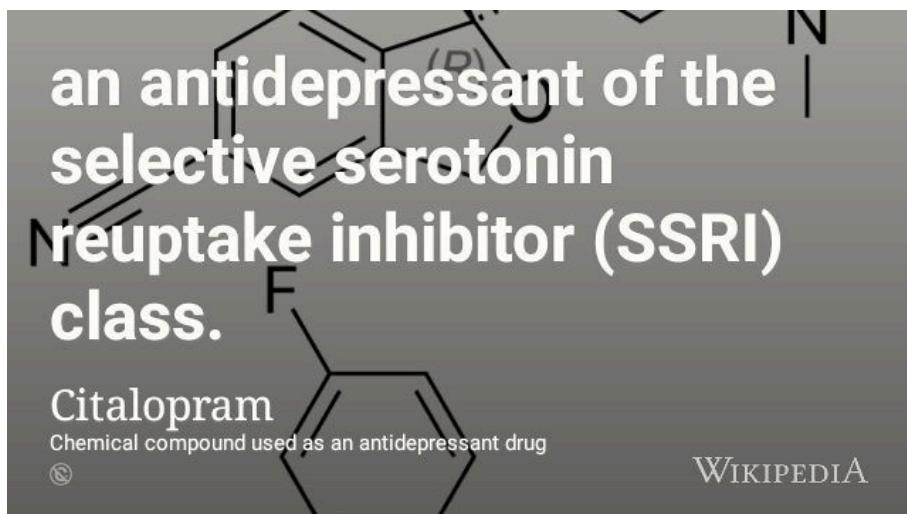


Figure 3.7: Citalopram is a type of antidepressant known as a Selective Serotonin Reuptake Inhibitor (SSRI). SSRI's can help some people who have been affected by depression. They work for some people (including me) but they don't for everybody. Skeletal formulae of Citalopram by Vaccinationist via Wikimedia Commons w.wiki/3Ddn adapted using the Wikipedia app.

Some doctors prescribe benzodiazepines for anxiety, which may be effective where SSRI's are not, but these can be addictive and have big side effects.

It is worth considering cognitive behavioural therapy (CBT) before taking any medication. *The Science of Wellbeing* (TSOWB) at coursera.org/learn/the-science-of-well-being is an easy way to access some CBT free online, see the signposts section (section 3.8) at the end of this chapter. (Santos, 2022b)

3.3.4 Suicide

If you're feeling suicidal, you don't have to struggle with difficult feelings alone. If you're suffering from emotional distress or struggling to cope a Samaritan can face your problems with you. Whatever you're going through, samaritans.org are available 24 hours a day, 365 days a year. They respond to around 10,000 calls for help in the UK every day. No judgement. No pressure. Call them free any time, from any phone (in the UK) on 116 123. It's important to talk to *somebody*, see figure 3.8.

Suicide is the biggest killer of under 35's in the UK. Like many young men, Ian Curtis in figure 3.8 tragically took his own life when he was just 23 years old.

"Strange as it may sound, it wasn't until after his death that we really listened to Ian's lyrics and clearly heard the inner turmoil in them."—
Bernard Sumner

Ian Curtis

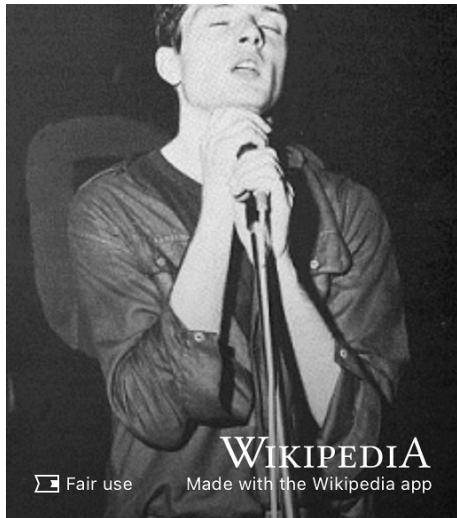


Figure 3.8: It might not be immediately obvious to you that people around you may be having suicidal thoughts. Ian Curtis was in a band called Joy Division, even his fellow band members like Bernard Sumner didn't realise how poor his mental health was until after his suicide. In retrospect it is obvious when you read his lyrics (Curtis et al., 1980). However, people who are affected by poor mental health often don't talk about it until it is too late - so it can be tricky to diagnose and help them in time. Fair use picture of Ian Curtis by Kevin Cummins on Wikimedia Commons w.wiki/4iWJ adapted using the Wikipedia app.

Globally, over 700,000 people die due to suicide every year, see who.int/health-topics/suicide. Every life lost to suicide is a tragedy because suicide is preventable, by talking and listening. If you're feeling suicidal or you know someone who is, don't suffer (or let others suffer) in silence. Check out the resources in section 3.6.4 and 3.6.5.

3.4 Look after yourself

Looking after yourself can serve to both prevent and treat mental health issues that can affect you in life. You might be your own worst critic, or perhaps when you're under pressure you neglect things that are proven to be beneficial for your mental health, like sleep, exercise, mindfulness and friendship. Looking after yourself means at least three things:

- being mindful of your feelings and learning to manage your inner critic
- being kind to yourself in various ways
- deliberately scheduling protected time to do the non-work things that matter.



Figure 3.9: It's important not to neglect your body, mind, soul and social life when you're working hard. Look after yourself by Visual Thinkery is licensed under CC-BY-ND

Harvard Psychologist Laurie Santos describes five evidence-based strategies for coping when times are really challenging and tough in the video in figure 3.10. Those strategies are:

1. **Exercise:** getting regular exercise improves both physical AND mental health.
2. **Gratitude:** research shows that being grateful can significantly improve

your mental health. One way to do this is by keeping a gratitude journal, a log you fill in everyday of things you are grateful for. This can include both the small and big things in life.

3. **Sleep:** actively developing healthier sleep patterns. Poor sleep hygiene can be both cause and effect of poor mental health. See the discussion of *Why we sleep* (Walker, 2018) in section 3.8
4. **Socialising:** give a higher priority to spending time with friends and family. There is good evidence to show this will make you happier than turning inward or diving deeper into work which could just make you more stressed, depressed or anxious (Marchese, 2022)
5. **Mindfulness:** be mindful of emotions using the R.A.I.N. technique:
 - **Recognise:** negative emotions
 - **Accept:** accept and allow emotions rather than fighting or ignoring them
 - **Investigate:** notice how the emotion feels inside your body
 - **Nurture:** be kind to yourself, step away from your emotions by distancing yourself from them.

It can help to think of negative emotions as coming from another person, an inner critic, rather than yourself. You are not your emotions and thoughts. Laurie explains the R.A.I.N. technique in figure 3.10.

The techniques described by Santos are not merely trashy pop psychology but scientifically proven techniques. (Yaden et al., 2021) So there are some simple things you can do to help yourself, but you may also need to seek help from others.

3.5 Burnout

Sometimes a desire to be productive by overworking has exactly the opposite effect, because the sacrifices you make for work can be counter-productive. If you burn too brightly you might burnout completely, see [mentalhealthuk.org/burnout](https://www.mentalhealthuk.org/burnout).

Sacrificing sleep, socialising, exercise and time-out to recharge your batteries can make you *less* productive, especially in the longer term.

I'm **not** saying don't work hard. Like Paul Nurse, I am arguing that you shouldn't work *too* hard:

"if you really want to be good you musn't work too hard. If you work too hard just hard all the time you will keep going in the same direction. One big advantage of taking some relaxation is it makes your brain come down and then you come and look at it freshly. When you take off the pressure, to go off and go and walk for day then you can imagine new things."

— Paul Nurse (Nurse, 2017)

Figure 3.10: Laurie Santos describes five coping techniques for improving well-being: Exercise, gratitude, sleep, getting social and meditation. You can also watch the six minute video embedded in this figure at youtu.be/kCQim9VrnDY (Santos, 2020).

So work hard but don't overwork. Overworking will literally kill you. (Ro, 2021)

If you're making too many sacrifices to get the grades you want, ask yourself, is it really worth it. See section 1.6, 2.4.1 and read Is a first class degree really that important? You may need to recalibrate your relationship and expectations of work to either recover from burnout, or stop burning out in the first place (Santos, 2022a)

3.6 Help is available if you need it

If you are affected by mental ill health, particularly anxiety or depression, it can be hard:

- to recognise that you need help in the first place
- to help yourself using self-help resources
- to ask others to help you

Even if you don't need help, it is important to recognise and understand the symptoms of mental ill health. It's quite likely that someone you know will suffer from mental health issues and as their friend or peer, it might be you that can help by encouraging them to get the help they wouldn't otherwise ask for.

You are not alone, help is available if you (or your friends) need it from a wide variety of sources:

3.6.1 Your University

There are lots of people who can help you:

- your personal tutor or other academic members of staff
- non-academic staff in the University, for example in Manchester contact the Student Support Office (SSO) studentsupport.manchester.ac.uk
- counselling services, for example contact counsellingservice.manchester.ac.uk. The counselling service offers help on dealing with anxiety, depression, exam stress, confidence and other issues.
- peers, flat-mates, family, friends etc. People close to you can help, although some people affected by mental health find it easier to discuss mental health with a trained professional or volunteer because of the social stigmas. There are lots of services outlined below that provide this kind of service.

3.6.2 The National Health Service

As a student studying in the UK you are entitled to access free healthcare provided by the National Health Service (NHS) of the United Kingdom. To do so you'll need to be registered with your general practitioner (GP), see nhs.uk: Getting medical care as a student

Your doctor can advise you on medical treatment if required, see for example [nhs.uk/conditions/antidepressants](https://www.nhs.uk/conditions/antidepressants)

3.6.3 Nightline

Nightline nightline.ac.uk is a confidential listening and information service run by students for students. Nightline is open 8pm till 8am every night during term time. It offers anonymous, non-judgmental and non-advisory support for students as described in figure 3.11.

Figure 3.11: Students explain in their own words how calling Nightline helped them whilst at university. You can also watch the 1 minute video embedded in this figure at youtu.be/RsfNkSZmUc8 (Association, 2016)

Manchester students can contact nightline at nightmail@nightline.manchester.ac.uk and expect a reply within 48 hours. See manchester.nightline.ac.uk for details.

3.6.4 The Samaritans

The Samaritans are a charity who provide emotional support to anyone in the United Kingdom and Ireland that:

- is suffering from emotional distress

- is struggling to cope
- is at risk of suicide

The name of the charity comes from the Parable of the Good Samaritan although the organisation itself is not religious. The Samaritans are available 24 hours a day, seven days a week, to talk confidentially about any problem, however big or small. See samaritans.org or telephone 116 123.

3.6.5 Papryus

Papyrus believe that many young suicides can be prevented, they are a national charity that you can find out more about at papyrus-uk.org or telephone the free number 0800 068 4141. See their application [stayalive.app](#)

3.6.6 Students Against Depression

Students Against Depression (SAD) acknowledge the devastating impact that depression can have on those experiencing it, as well as on their friends, family and supporters. For further help in understanding and coping with suicidal thoughts, and emergency contacts in a crisis, visit studentsagainstdepression.org

Actor Ruby Wax has written about mental health and how the “internal critics” in our minds can send us mad in her book *Sane New World*. (Wax, 2014b) She is interviewed by Students Against Depression in the video in figure 3.12 about using mindfulness to “dodge the bullets” of depression.

3.6.7 Self-help services

Self-Help services are a mental health charity which helps people to help themselves, see selfhelpservices.org.uk or phone 0161 226 3871.

3.6.8 MIND

MIND provide advice and support to empower anyone experiencing a mental health problem. They campaign to improve services, raise awareness and promote understanding of mental health issues. Find out more at mind.org.uk and in the video in figure 3.13

3.6.9 Student minds

Student Minds empowers students to look after their own mental health, support others and create change, find out more at studentminds.org.uk and in the video in Figure 3.14 which describes why it is important to talk about student mental health.

Figure 3.12: Ruby Wax describes being affected by depression in her childhood and how mindfulness and cognitive behavioural therapy (CBT) provided an alternative to medical treatment enabling her to dodge the bullets of mental health. You can also watch the 2 minute video embedded in this figure at youtu.be/7TCCLY_lV6M (Wax, 2014a)

Figure 3.13: Stephen Fry, President of Mind, describes how MIND tackles misconceptions around mental health and social stigmas. You can also watch the four minute minute video embedded in this figure at youtu.be/OPxHTZRwXX8 (Fry, 2014)

Figure 3.14: Talking about mental health is a crucial part of helping those who are suffering from it. You can also watch the 1 minute video embedded in this figure is also available at youtu.be/PiqcSSDwvvY (Minds, 2016)

3.6.10 Togetherall

Togetherall is an online community for people who are stressed, anxious or feeling low. The service has an active forum with round-the-clock support from trained professionals. You can talk anonymously to other members and take part in group or 1-to-1 therapy with therapists. Togetherall is for anyone aged 16 or over who wants to improve their mental health. The service is free for many universities. Find out more at togetherall.com and in the video in figure 3.15 which describes why its important to talk about student mental health.

Figure 3.15: A quick look inside togetherall, an online community for people who are stressed, anxious or feeling low. You can also watch the 1 minute video embedded in this figure at youtu.be/mXEbpau9YjU (togetherall, 2020)

3.7 Developing a growth mindset

Learning at University can be hard because you might have gone from being at (or near) the top of the class in high school to no longer being top of the class at University.

Likewise the job hunting described in chapter 8 can take a heavy toll on your mental health because repeated rejection is an ordinary part of the process. It can be time consuming, stressful and demoralising. You may find your applica-

tions disappear into a black hole. They will be ghosted (ignored) by employers. Interviewers will blank you and refuse to give you meaningful feedback because they're too busy. This could happen multiple times. This is all *par for the course*, normal and expected, and is not necessarily a reflection on your abilities or potential. See the examples in the coding interview section 10.2.3. (Davidova, 2021)



Figure 3.16: A fixed mindset is monolithic like the Easter island statues (mo ai). If you’re not already, you should be wary of a fixed mindset. Fixed mindsets by Visual Thinkery is licensed under CC-BY-ND

Adopting a growth mindset can be a successful strategy for maintaining your wellbeing, see figure 3.16. If your grades aren’t as good as you hoped or your search for employment is being met with repeated rejection, a growth mindset can help. Let’s take rejection from potential employers as an example, there are two ways you can react to it:

1. **Fixed mindset:** responding with a fixed mindset will mean you are likely to take rejection personally. You might even assume that this confirms what you’ve always suspected. You’re not good enough or that you made some fatal mistake in your applications or interviews. Ouch.
2. **Growth mindset:** by responding to rejection with a growth mindset, you focus on what happens next. Rejection is not failure but a “not yet” described in figure 3.17. Maybe you’re not yet ready for that employer, but you’ll definitely have a good idea of what you learned from the process and how you can do better next time.

According to Stanford psychologist Carol Dweck we can all be placed on a spectrum describing where we think our abilities come from. At one end, the fixed mindset assumes all kinds of abilities are fixed traits while at the other end, a growth mindset assumes these abilities can be developed over time. (Dweck,

2017) There is good evidence that adopting a growth mindset will make you a better learner who can cope with the inevitable failures and rejections in life better. This approach can be used in a range of different disciplines such as learning programming languages (Cutts et al., 2010), music (Davis, 2016) and job hunting.

Figure 3.17: Psychologist Carol Dweck explains the power of *not yet* and the growth mindset. You can also watch the 11 minute video embedded in this figure at youtu.be/J-swZaKN2Ic (Dweck, 2014)

3.8 Wellbeing signposts

This chapter has looked at your wellbeing and especially the role that both your mental health and physical health play in your future. Matt Haig's first-hand accounts of poor mental health will be comforting to anyone who is affected by mental ill health. Even if you're not affected, there is a 25% chance you will be at some point in your life. There's also a high probability someone close to you will suffer from mental health issues. It might be a colleague, friend, family member, fellow student or partner, so it is worth educating yourself on the issues by reading his two short books:

1. *Notes on a Nervous Planet* is a personal account of anxiety (Haig, 2019)
2. *Reasons to Stay Alive* is a personal account of depression (Haig, 2016)

What's good about Matt Haig's books is they are quick and easy to read, but give plenty of first-hand insight into what mental ill-health can do to people (including you). Matt describes his top five tips for good mental health in figure 3.18

Figure 3.18: Two of Matt Haig's top five tips for good mental health include 1. Being more careful (and mindful) of social media and 2. Reading more books because books are good for your soul. Not just his book. Any book. Books are good for you. Trust me on this. You can also watch the 4 minute video embedded in this figure at youtu.be/JMm5kiiMjTM (Haig, 2018)

There's plenty of evidence that social media can have a detrimental effect on health. Jaron Lanier's skeptical polemic *Ten Arguments for Deleting Your Social Media Accounts Right Now* (Lanier, 2018) is a thought-provoking romp through some of the pitfalls of social media that may have you reaching for the delete or un-install button fairly quickly. You don't have to be on social media, see figure 3.19.

If all these books are making you sleepy, neuroscientist Matthew Walker's *Why We Sleep: The New Science of Sleep and Dreams* may change your view on the importance of a good nights sleep. (Walker, 2018)

Finally, it's well worth taking a look at Laurie Santos course on *The Science of Wellbeing* (TSOWB) at coursera.org/learn/the-science-of-well-being. (Santos,



Figure 3.19: Social media like LinkedIn clearly has its uses (see section 6.7.5) but you don't *have* to be on it at all. Poor mental health is just one reason to be wary of social media, for nine other reasons read Jaron Lanier's polemic. (Lanier, 2018) Notes to Strangers by andy-leek.com photographed by Duncan Cumming

2022b) TSOWB course provides an alternative to medication as it follows the principles of cognitive behavioural therapy (CBT).

TSOWB is the most popular course at Yale University and looks at some simple techniques you can use to improve your happiness. (Shimer, 2018) The course will help you increase your happiness and build more productive habits. Using the latest research, Santos describes the misconceptions about happiness and “annoying features” of your mind that can impair your well-being. The course takes about 19 hours to complete but you can spread this over a whole semester (or longer) if you choose. The short clip in figure 3.10 gives you a brief taster of Laurie’s style and work.

3.9 Breakpoints

Let’s pause here. Insert a breakpoint in your `code` and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

- How would you describe your own state of mental health?
- Do you have friends or peers who are affected by mental ill health?
 - What are the signs they might be suffering?
 - How could you support or help them better?
- If you describe your own mental health as poor
 - Where can you go for self-help?

– You are not alone but who can you talk to?

* RESUME

3.10 Summarising well-being

Too long, didn't read (TL;DR)? Here's a summary:

Anxiety and depression are serious conditions that are very likely to affect you or somebody close to you while you are at University. There's a one in four chance that you will be affected by mental health issues at some point in your life.

We've only talked about two particular mental health issues, anxiety and depression, but there are many other conditions such as phobias, obsessive-compulsive disorder (OCD), eating disorders, self-harm and more that are beyond the scope of this chapter. They do have one thing in common, and that is that talking about them is an important part of starting to develop better mental health. Mental health can affect everyone, all genders, classes and races.

If you are affected by mental ill health, talking about it is the best place to start, but often the hardest. You can continue by:

1. Accepting yourself for who you are, see chapter 2
2. Learning to identify and challenge unhelpful thinking patterns
3. Avoiding the trap of compare yourself to other people too much

... are three more steps you can take to help yourself, see figure 3.20. In this chapter, I've outlined some ways you can help yourself alongside some of the services and people you can talk to if you need to.

Despite how you might feel, you are not alone.

Nurturing your future is Coding your future.

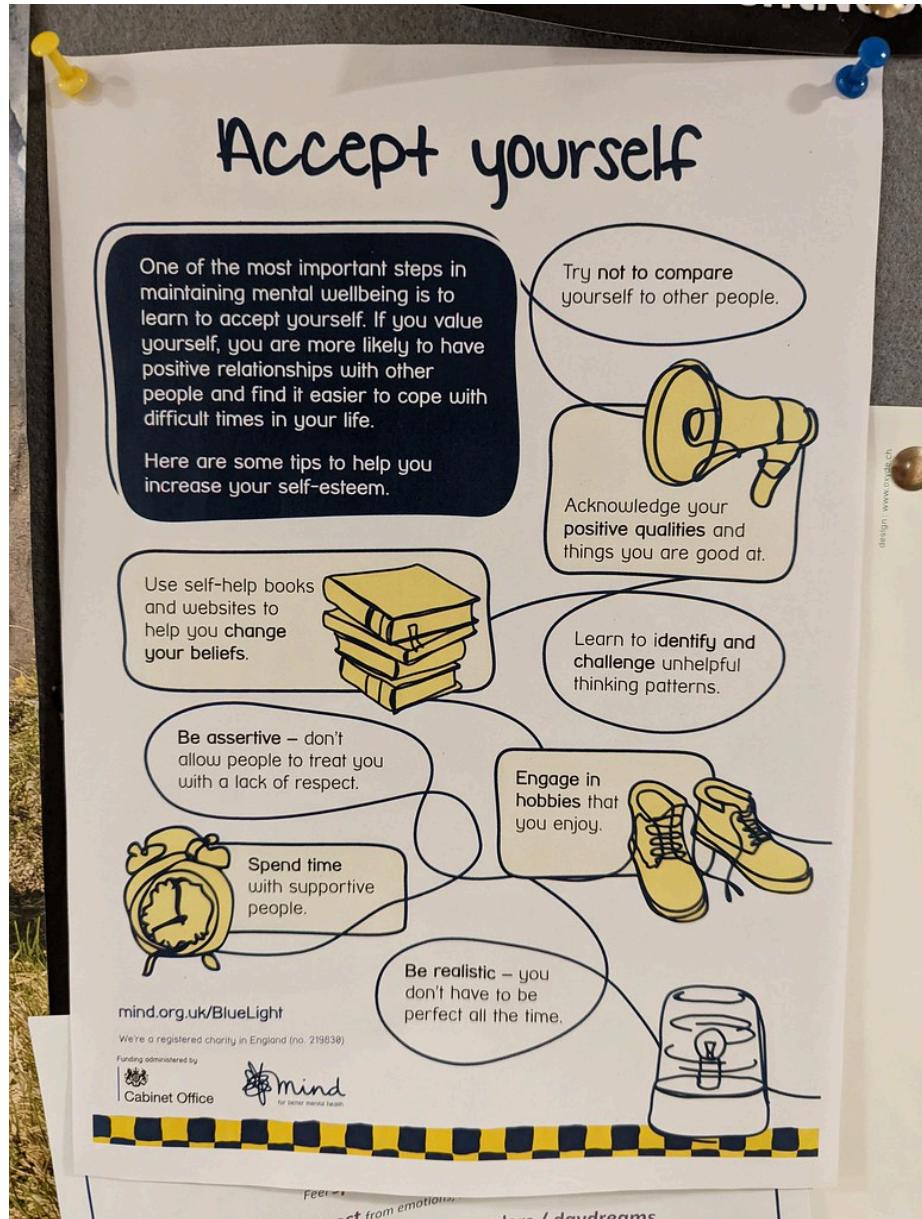


Figure 3.20: Accepting yourself for who you are will improve your mental health and self-esteem. This can, in turn, help you prevent or treat problems caused by anxiety and depression. Poster via mind.org.uk

Chapter 4

Writing your future

We all have strengths and weaknesses, and for students of Computer Science, it is often their communication skills that let them down. Other chapters in this guidebook discuss the fundamental communication skills of speaking, listening and reading. This chapter looks at *written* communication because:

- every *engineer* is also a writer, and good engineers have good written communication skills
- every *scientist* is also a writer, and good scientists also have good written communication skills too

Your ability to write clearly isn't just important when putting together CV's like in figure 4.1. Your ability to write clearly isn't just important when composing covering letters or filling in application forms either, it's a fundamental skill that will help you become a valued employee in the longer term. Learn to write more clearly now and you'll improve your chances of success later. Learn how to get better at writing your future.

4.1 What you will learn

In this chapter you will learn to:

1. Recognise the importance of written communication, both as a reader and a writer
2. Identify examples of where written communication is crucial in teams of scientists and engineers
3. Improve your written communication skills using some simple writing and reading exercises



Figure 4.1: There are many motivations for studying computer science, unfortunately for some students, minimising the amount of reading and writing they have to do in natural languages (like English) can be one of them. You can *run* from written communication, but you can't *hide* from it. Writing a CV, covering letter or filling in an application form is just the tip of the *writing iceberg*, section 4.4.2 outlines the some of the rest. This chapter looks at some simple techniques for improving your written communication skills. Avoidance sketch by Visual Thinkery is licensed under CC-BY-ND

4.2 Writing as a soft social skill

Your soft social skills will take a **life time** to develop and are **really hard** use. Why? Because soft skills are about *communicating* with and *understanding* other people so that you can work *together* as a team toward a shared goal. Your soft skills are hard.

There are very few jobs where you work on your own in complete isolation. For example, most software and hardware is designed, built, tested and used by teams of people. Many of these teams are large and have very diverse membership. This means that sooner or later you're going to have to master the dark arts of *working with other people* by developing and deploying your softer skills. One of those softer skills you'll need to continuously develop is written communication.

Communicating with other people and working in teams is inherently difficult because we're all human. There is good news and bad news...

- **THE GOOD NEWS** is, people can be diligent, humble, competent, honest, caring and reliable. They can be co-operative, generous, supportive, kind, thoughtful, intelligent, sensitive, understanding, punctual and professional too!
- **THE BAD NEWS** is, unfortunately people can also be lazy, stupid, ignorant, vain, incompetent, dishonest, unreliable, greedy, egomaniacal, unpredictable and moody. They can be proud, selfish, competitive, lustful, angry, envious, mean, busy, insensitive and thoughtless too. Some will disagree with you, boss you around, betray, exploit, misunderstand and mislead you, deliberately or otherwise. (Goble, 2007)

This *shouldn't* be news to you but it means communicating with and understanding other people can be hard work, but don't worry, **everyone** finds this challenging, it's not just you! It doesn't matter if you're an extrovert or an introvert, communication is a challenge for everybody, and everyone can get better at it too. It's easy to undervalue the importance of communication because we do it everyday. Scientists and engineers can be particularly bad at this, often overlooking the importance of written and spoken language. It probably doesn't help that in the UK, and many other countries, many students in the latter stages of their education, choose either

- an exclusively scientific-mathematical path OR
- an "arty" humanities path

This means that the two cultures of humanities and science are thriving, but still living in separate houses like an estranged and bickering couple. (Snow, 1959) In the worst case scenario, two cultures in society produces:

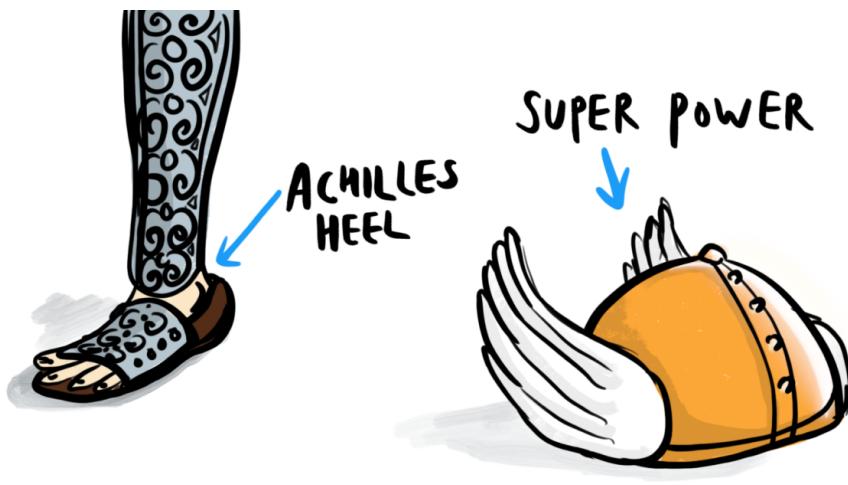
1. graduate scientists and engineers with weaker communication and literacy skills

2. articulate and literate humanities graduates with weaker technical & numeracy skills

Are you in the first category? This chapter takes a look at the softer skill of written communication and techniques you can use to improve your writing. Whatever mood your readers are in, they'll find it a lot easier to work with you when you can express yourself clearly in writing.

4.3 Computing is your superpower!

Studying computer science gives you an *awesome* superpower. We will look at some of the reasons why in chapter 5 on *Computing your Future*. But for now, let us just acknowledge that hard technical skills like computing are highly sought after and valuable, both commercially and otherwise.



© VISUAL THINKERY

Figure 4.2: Computing is a superpower that gods like Hermes and mortal heroes like Achilles would probably have envied. (Fry, 2018, 2017) As a technical or *hard skill*, computing is a crucial weapon in your armoury but what are your weaker skills? What is your Achilles' heel? For some scientists and engineers, their weakness is their soft skills, such as communication. Writing, reading, speaking and listening to communicate with team members can be a weaker skill for many technical people. This chapter looks at what you can do to improve these skills and convince employers that you are rounded individual with a healthy balance of soft and hard skills. Achilles heel to superpower by Visual Thinkery is licensed under CC-BY-ND

Your computational superpower is less powerful if it isn't complemented by a broad range of softer skills. Typically, these skills are not closely examined in most computer science degrees, for example by repeated assessment. There's an

awful lot of technical computing curriculum that tends to have a higher priority over soft skills. It's not that soft skills aren't important but that they can be hard to quantify.

For example, if I want to know how good you are at understanding the syntax and semantics of a programming language like Python, there are tried and tested techniques for measuring this. However, if I want to know how good you are at using your communication skills to work in a team of engineers to negotiate, lead, resolve conflicts, persuade others, show empathy etc that's **much** harder to measure accurately. What technical degrees tend to measure is technical skills because softer skills, shown in figure 4.3, are much harder to assess by examination and coursework. This is one of the many reasons that you are much more than your grades, see figure 12.3.

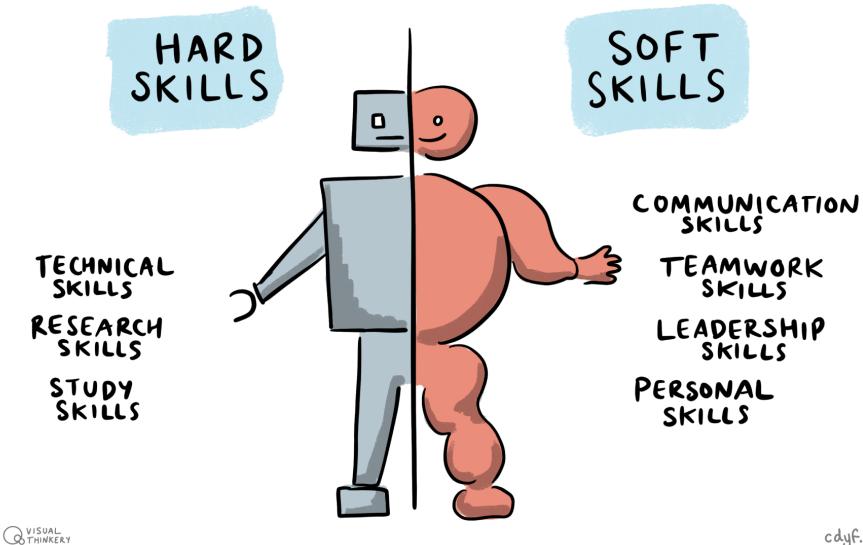


Figure 4.3: Hard skills and soft skills aren't much use without each other. You will need both to survive and thrive but most science and engineering education focuses on your hard skills, not your soft skills. Why? Because hard skills are often much easier to measure. Hard and soft skills sketch by Visual Thinkery is licensed under CC-BY-ND

Let's look at some of low-level communication skills (I/O) that they are built on.

4.4 Communication I/O

In terms of input and output, your fundamental communication skills are listening, speaking, reading and writing words in natural languages shown in table

Table 4.1: The inputs and outputs of the fundamental assembly languages of human communication

	Input	Output
Written natural language	Reading	Writing
Spoken natural language	Listening	Speaking
Nonverbal language	Observing other people	Being observed by others

4.1. These are the “assembly languages” of human communication. This might sound blindingly obvious, but these skills are often under-estimated or under-valued by engineers and scientists, especially undergraduates. Alongside verbal and written communication, there’s also nonverbal language, or body language such as eye contact, gestures and facial expressions.

Some people leave plenty of room for improvement when it comes to the communication skills outlined in table 4.1. Think of:

- the stereotypical mad scientist, clad in a white coat, unable to explain the complexities of their research to people inside their lab, let alone outside of it, see figure 4.4
- the nerdy software engineer stereotype who prefers the company of computers to people

Yes, these are lazy tropes and unhelpful stereotypes, but they express public perception of scientists and engineers as poor communicators. Don’t perpetuate the stereotype by being a bad communicator.

4.4.1 The pen is mightier than the sword

The art of communication is a huge subject which extends far beyond the scope of this guidebook. So for the rest of this chapter, we’ll focus on your superpower of written communication. The pen (and keyboard) are mightier than the sword, see figure 4.5.

Written communication skills are important because:

1. **Good writing and reading are crucial in applications** for employment and further study. From writing CV’s, covering letters, completing application forms and reading job specifications, and employer (or course) information, your ability to read and write in natural languages is crucial to coding your future.
2. **Writing often gets neglected:** Written communication skills (both reading and writing) are sometimes sidelined in science and engineering degrees. This is particularly true in the “hard sciences”. For example, communicating and solving problems using code or mathematics are usually the dominant forms of assessment in computer science courses. That’s un-

described as "mad" or "insane" owing to a combination of unusual or unsettling personality traits and the unabashedly ambitious, taboo or hubristic nature of their experiments

Mad scientist

 CC BY-SA

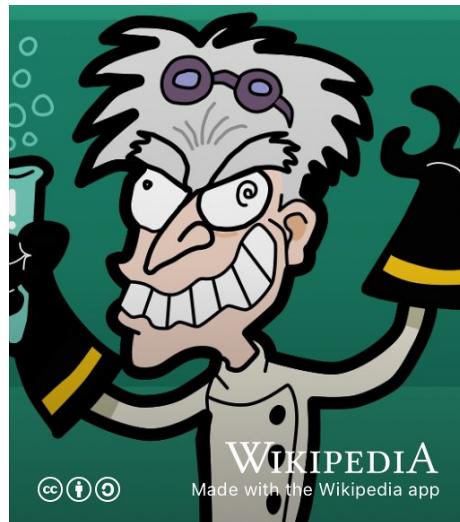


Figure 4.4: Are you mad, bad and dangerous to know? You'll be familiar with the mad scientist stereotype: perceived as an evil genius owing to a combination of unsettling personality traits and the unabashedly ambitious, taboo or hubristic nature of their experiments. Besides their supposed insanity, another reason mad scientists (and engineers) are perceived badly, is their poor communication skills. Maybe if they could communicate better the public might understand and trust them more? Don't perpetuate the stereotype by being a poor communicator, there are too many of those already! CC BY-SA cartoon of a mad scientist by J.J. on Wikimedia Commons w.wiki/5JeU adapted using the Wikipedia App

"The pen is mightier than the sword" is a metonymic adage, created by English author Edward Bulwer-Lytton in 1839, indicating that the written word is a more effective tool for communication than violence

The pen is mightier than the sword



Figure 4.5: Do you need more weaponry in your life? Besides computing, you should arm yourself with communication superpowers. It's in your own selfish interests to continuously develop your written communication skills because the pen is mightier than the sword. Public domain image of a drawing of Cardinal Richelieu by Henry Alexander Ogden via Wikimedia Commons w.wiki/3WHg adapted using the Wikipedia app

derstandable given the subject, but tends to push natural languages (like english) to the sidelines.

3. **Good engineers are also good writers** Many engineers (and scientists) could significantly improve their written communication skills. Software engineers are notoriously bad at writing, see for example Why Computer Science Students Need Language, (Beaubouef, 2003) *Scientists Must Write* (Barrass, 2002) and The Real Reason Silicon Valley Coders Write Bad Software, (Meisler, 2012) just three examples amongst many others making exactly the same point. Employers like Google provide training (and a whole career path) for technical writers, see developers.google.com/tech-writing. I'm glad they exist, but these careers and courses wouldn't be needed if software engineers were better at documenting, explaining and communicating with other human beings in the first place!
4. **Writing good english is like writing good code.** Some of the skills you already have in coding can be transferred to written communication. Just like a good `function`, `class` or `method` in code should be well-defined with a clear purpose, your writing should also be clear and coherent. Well structured writing is a lot like well architected software too, with a clear separation of concerns (SoC)
5. **It is relatively easy to improve** your written communication skills, simply by reading and writing more. Reading and writing deliberately every day, will significantly improve these skills. See the rest of this chapter for some simple exercises to get started with and:
 - chapter 6 on *Debugging your future*
 - chapter 15 on *Hacking your future* for details.

4.4.2 Natural language engineering

As a species, we've been writing stuff down for millenia in order to communicate with each other, see figure 4.6. If you stop to think about it, engineers and scientists spend a *lot* time communicating in writing. As well as engineering `code`, they also spend a significant amount of time engineering *messages* in natural languages like english.

Consider the following:

- email and instant messaging, Slack, Microsoft Teams, Discord, Zoom etc
- Posting on social media: LinkedIn, Facebook, WhatsApp, Twitter, blogs, Medium.com etc
- bug reports and messages in issue trackers like Jira, BugZilla, Trello and version control systems like Github and Gitlab etc
- ‘How to’ and cookbook style articles and books
- API reference material
- in-code documentation `# comments in code`
- Self-documenting code that describes itself
- Executable specifications in test suites like cucumber.io
- Laboratory manuals and laboratory notebooks

**“Did you ever notice
that, when people
become serious about
communication, they
want it in writing?”**

Writing

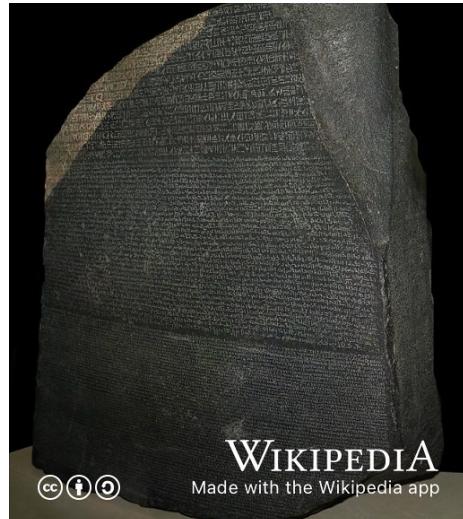


Figure 4.6: Have you ever noticed how when people become serious about communication, they want it in writing? CC BY-SA image of language in Ancient Egyptian using hieroglyphic, demotic and Ancient Greek written on the Rosetta Stone by Hans Hillewaert on Wikimedia commons w.wiki/3Ycn adapted using the Wikipedia app

- The one or two page summary for management
- reddit.com and hacker news news.ycombinator.com etc
- User documentation, release notes
- Case studies of software use
- Frequently Asked Questions (FAQ)
- Personal websites YourPersonalDomain.com if you have one
- Questions and answers on forums like stackoverflow.com
- Commit messages in version control systems like git and mercurial etc
- Architecture documentation and design specifications
- Literate programming natural language descriptions of computational logic (Knuth, 1984)
- Jupyter.org notebooks, code and natural language mixed together, as do many other systems like, quarto.org and bookdown.org

What do they all have in common? They're all written in natural languages like the English language, but without them, the software or hardware they describe and discuss would be useless. Using written (and spoken) natural language is a key social skill and a communication skill. You might be using your language to influence, persuade, convince or argue with members of your team. So it is important that you do it to the best of your abilities. Computer Scientist Luis von Ahn, shown in figure 4.7, is the creator of Duolingo.com. Luis regrets that he didn't spend any effort developing his social skills early in his career. Instead,

like many computer scientists he focussed his effort on his mathematical and technical skills instead. (Shaw, 2021)

**"As a CEO one thing I
wish I'd concentrated on
earlier was my social
skills"**

Luis von Ahn



Figure 4.7: Luis von Ahn is the Chief Executive Officer (CEO) of Duolingo. Since 2012, the duolingo app has been downloaded 500 million times. Luis wishes he'd concentrated on developing his social skills earlier in his career. Are you developing your hard technical skills at the expense of your softer social skills? Portrait of Luis von Ahn by EneasMx on Wikimedia commons w.wiki/4VPU adapted using the Wikipedia app

Making software isn't all about what you can do as an individual but rather how you communicate with and contribute to your team. It's easy to get carried away with your ego. Remember that most jobs require *lots* of softer people skills and collaboration, written communication is a huge part of that, alongside spoken communication see for example The Myth of the Genius Programmer. (Fitzpatrick and Collins-Sussman, 2009)

4.4.3 Writing as therapy

Besides being a tool for communication, writing can be therapeutic for the writer too. The simple act of writing can help you tackle mental health issues discussed in chapter 3. Prime minister Winston Churchill found that writing helped to keep the "black dog" of his depression under control, see section 3.3.2 and figure 4.8.

So writing can be therapeutic, a powerful tool for developing mindfulness as well as communicating with other people.

Writing was his main safeguard against recurring depression, which he referred to as his "black dog"

Winston Churchill

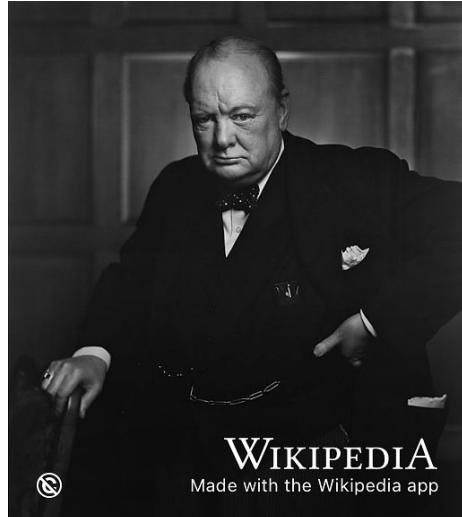


Figure 4.8: Winston Churchill managed to regulate his depression by writing. If you are affected by anxiety or depression, you may find that private forms of writing like those described in section 4.5.3 will help to improve your mental health. Public domain portrait of Winston Churchill by Yousuf Karsh on Wikimedia Commons [w.wiki/4aJk](https://commons.wikimedia.org/w/index.php?title=Winston_Churchill_by_Yousuf_Karsh&oldid=10231110)

4.5 Writing your future

Hopefully I've convinced you that written communication skills (both as a writer and reader) are important soft skills that engineers often neglect. So how can you improve?

4.5.1 Dogfooding

Many employers test their products and services by trialling them on their own employees, this is known as eating your own dogfood shown in figure 4.9. Tasty, tasty dogfood.

Dogfooding is a simple technique for testing your own writing. Let's say you've just written a personal statement or covering letter (see section 6.10). It's natural to read it over in your head to check for errors, before you send it. However, **reading it aloud** will pick up errors you may not have spotted by reading silently. Your ears and tongue will pick up on awkward phrases and sentences that your eyes didn't spot. There's something about articulating words out loud that flushes out errors you don't pick up when you read them in your head. Dogfooding is a tried and tested technique. It also means you're ready to vocalise those answers in an interview.

You might want to choose carefully where you do this as it might look a bit



Figure 4.9: Reading your own writing (aloud) is like eating your own dog food. It's a simple and proven technique for improving your written communication in job applications such as covering letters, CVs, personal statements and the like. Dogfooding by Visual Thinkery is licensed under CC-BY-ND)

strange, but it works well. If you talk into a mobile phone while looking at a piece of paper, people won't notice you're talking to yourself. But you'll probably need some privacy as the stuff you're talking about is likely to be personal. Alternatively, find a critical friend who you can read out loud to. You can also use a screen reader, for example :

- Microsoft Teams has an immersive reader
- Android devices have TalkBack
- Apple devices have VoiceOver

A screen reader will often reveal mistakes you haven't spotted by reading your text silently.

4.5.2 Try Google's Tech Writing course

Google have developed some excellent written communication courses for software engineers, and those looking to become technical writers:

1. Technical Writing One: Technical Writing Fundamentals for Engineers
developers.google.com/tech-writing/one
2. Technical Writing Two: Intermediate Technical Writing for Engineers
developers.google.com/tech-writing/two

You might also be interested in the follow on courses:

1. Tech Writing for Accessibility developers.google.com/tech-writing/accessibility
2. Writing Helpful Error Messages developers.google.com/tech-writing/error-messages

The first two tech writing courses run as part of the second year course COMP2CARS at the University of Manchester, see chapter 19 for details.

Google occasionally delivers these technical writing courses as free sessions open to the general public. For details, see developers.google.com/tech-writing/announcements for details.

4.5.3 Deliberate daily writing

Another technique for improving your written communication is to write something every day, that might be a personal diary that only you read, or it could be something more public like blog. Schedule a time every day, say 15 to 30 minutes when you will do this without fail. That writing could take several forms:

- public web log or blog
- gratitude journal see section 3.4
- private diary or personal laboratory notebook
- morning pages, a unedited stream of consciousness that can help you become more creative (Cameron, 1992; Burkeman, 2014)
- bullet journal. Some people swear by it, see bulletjournal.com

The technique of *30 minutes per day* can be a very effective way of getting things done, incrementally over time. In my experience it works for lots of things besides writing including getting exercise (discussed in chapter 3) to gardening. (Leendertz, 2006)

4.5.4 Deliberate daily reading

Reading other people's code will improve your software engineering skills. Likewise, reading other peoples writing will improve your natural language engineering skills. Read anything, it might be novels, magazines, newspapers, stuff online or any of the books cited in chapter 27. Find a time and place to do this every day and stick to it, see figure 4.10

4.5.5 Reading the friendly manual

You don't get good at communicating with computers (coding) by just *writing* lots of code. You also need to *read* other people's code too and be able to understand and modify it. Likewise, you don't get good at communicating with people by just *writing* stuff in natural languages like english. You need to *read* stuff too. Books, manuals, software documentation, articles, use cases, novels, poetry, plays, magazines, newspapers etc. Reading this stuff will help you learn and you'll improve your written communication skills too. So *Read The Friendly*

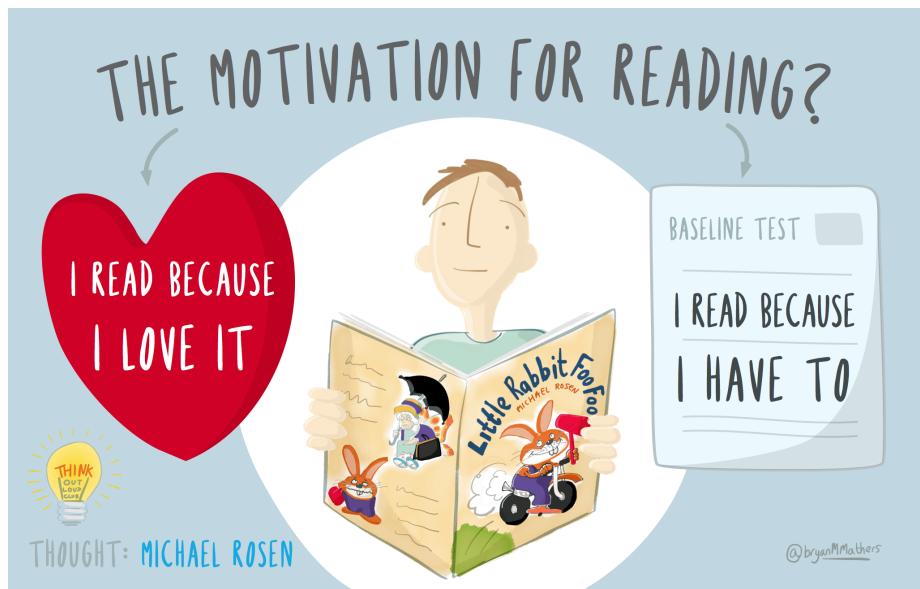


Figure 4.10: Reading allows you to learn from other people's hard won experience whilst also improving your own written communication skills. Just like you improve your coding skills by reading *and* writing code, you will improve your written communication skills by reading *and* writing in natural languages. The motivation for reading by Visual Thinkery is licenced under CC-BY-ND with help from Michael Rosen

Manual. RTFM. Read THIS Friendly Manual and the stuff it cites, see figure 4.11

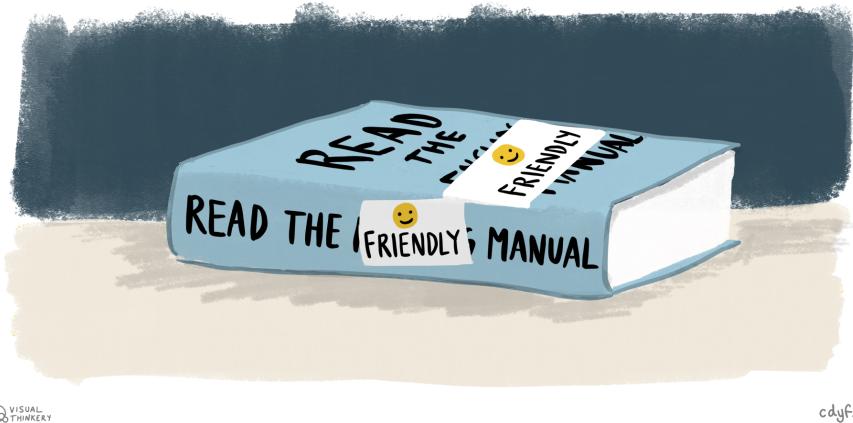


Figure 4.11: Read The Friendly Manual (RTFM), some of it you will love, some of it you won't. Either way reading will help you develop valuable skills and knowledge. Read The *Friendly* Manual by Visual Thinkery is licenced under CC-BY-ND

4.6 Breakpoints

Let's pause here. Insert a breakpoint in your `code` and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

- Which of the communication skills in table 4.1 are your strongest?
- Which of the communication skills in table 4.1 are your weakest?
- What activities could you do to improve your weaker communication skills?

* RESUME

4.6.1 Coding challenges

- Write an article or blog post about something you care about, find a suitable venue for publication
- Take a course from outside computer science, where the main form of assessment is written essays or dissertations. Humanities departments are a good place to start. This will improve your written communication skills

- Not been reading many books lately? Pick a book to read just because its interesting, rather than because you have to.

4.7 Summarising your soft skills

Too long, didn't read (TL;DR)? Here's a summary:

You'll need both soft and hard skills to compete in the workplace. Don't underestimate the importance of softer skills, we've looked briefly at written communication skills in this chapter but that's only the tip of the soft skills iceberg.

Teamwork, negotiation, conflict resolution, public speaking, motivating others and leadership are also important soft skills too. How can you develop these skills while at University? How can you demonstrate to potential employers that you have these skills? Your technical skills are of limited use without people skills, to allow you to work with others see figure 4.12.

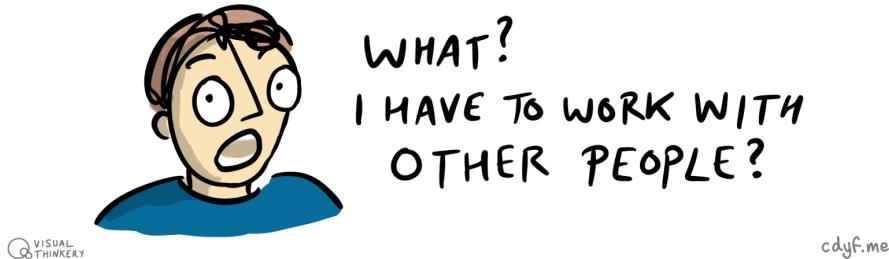


Figure 4.12: Unless you want to be a lighthouse keeper on a remote island, there are very few jobs where you don't have to work as part of a team with other people. If you were hoping to carry on working independently from other people, sorry to break the bad news! In the longer term, you will benefit from constantly improving your softer skills and providing evidence of them to potential employers. Written communication is just one of those softer skills. Other people sketch by Visual Thinkery is licensed under CC-BY-ND

So communicating with other people is a key skill, see figure 4.13. This chapter has looked at written communication, in chapter 10 we will look at spoken communication. Writing your future is coding your future.

This chapter is under construction because I'm using agile book development methods, see figure 4.14.



Figure 4.13: Your network can help you before, during and after your employment, so be sure to grow your network when you can. This includes both the stronger ties of trusted friends alongside the weaker ties too, described in section 8.2.5. Both are important. It's all about people by Visual Thinkery is licensed under CC-BY-ND

The Death Star is a fictional mobile space station and galactic superweapon featured in the Star Wars space-opera franchise

Death Star

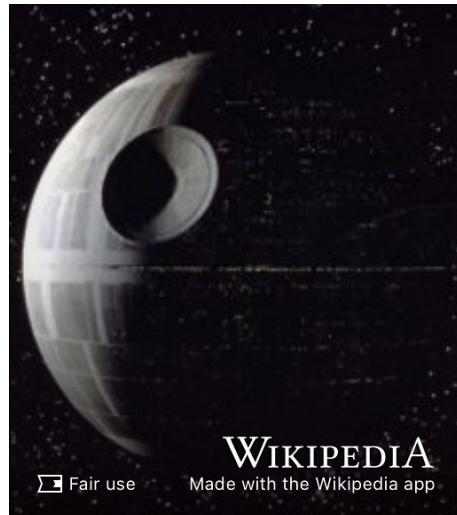


Figure 4.14: Just like the Death Star, this ~~galactic superweapon~~ book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Chapter 5

Computing your future

It's difficult to think of any aspect of our lives that hasn't been changed by the invention of the digital programmable computer, just 74 short years ago. (Lavington, 1998) Consequently, computing is a crucial skill in a wide range of careers across every sector of business and society. You don't have to have studied Computer Science at University to take advantage of all the exciting opportunities provided by computing. This chapter looks at why computing is a subject for everyone. If you're studying computing, this chapter isn't aimed at you, unless you are struggling to stay motivated with your subject!

5.1 What you will learn

Reading this chapter and doing the activities will help you to:

1. Identify where you can get started with computing, if you're not studying computer science as a major part of your degree
2. Describe why NOT studying computer science doesn't necessarily "lock you out" of computing as a career

But why should everyone be studying computing? There are social and economic arguments:

5.2 Computing is for everybody

At school, everyone learns to read, write and do maths. These are sometimes known as the three Rs but:

- Why did you learn to read and write? Was it so that you could become a professional writer?
- Why did you study mathematics? Was it so that you could become a professional mathematician?



Figure 5.1: Computing is much more than coding, this chapter looks at what computing can do for your future. CV work sketch by Visual Thinkery is licensed under CC-BY-ND

Of course not, that would be ludicrous! You learned to read and write because they are fundamental tools for expressing yourself and communicating with other people. You studied maths so that you could develop numeracy, reason about the world around you, analyse data and solve problems.

So why should everyone learn about computing? Is it so that everyone can become software engineers? Again, this is patently ludicrous.

Everyone should study computing for the same reasons everyone studies maths and english at school. Like writing, computing is one of the most creative tools for expression and communication that we have today. Just like mathematics, studying computing will also help you to solve important problems too. Sam Aaron, creator of Sonic Pi, puts exactly this case for creative computing in his TEDx talk (Aaron, 2016) shown in figure 5.2.

Figure 5.2: Sam Aaron puts the creative case for computing by discussing programming as performance in his TEDx talk. You can watch the watch the 18 minute video embedded in this figure at youtu.be/0lTZ8Tuyu5I. (Aaron, 2016)

Computing is also an intellectually stimulating and challenging subject to study in its own right. If you don't believe me, I'm not going to make the case here. If you are a humanities student, have a look at Silvio Peroni's free computational thinking and programming textbook at comp-think.github.io. (Peroni, 2021) If

you like doing Massive Open Online Courses (MOOCs), you might also enjoy CS50, with David Malan shown in figure 5.3.

Figure 5.3: David Malan on CS50, Harvard University’s introduction to the intellectual enterprises of computer science and the art of programming. You can also watch the full 2 minute introduction video embedded in this figure at youtu.be/3oFAJtFE8YU. (Malan, 2018, 2010)

5.3 Software is eating your future

Whatever future world you enter into after you graduate, there’s a good chance it has already been eaten by software. In 2011, the software engineer and billionaire investor Marc Andreessen outlined why (in his opinion, figure 5.4) software is eating the world, in *The Wall Street Journal* (Andreessen, 2011).

Unfortunately, many people lack the digital skills required to take advantage of all the opportunities provided by software and computing. Robert Sedgwick at Princeton University has, like many others, argued that Computer Science should be a **required** topic of study for *all* undergraduate students in University. (Sedgwick, 2019) We’re not there yet because computing is a subject that has historically been siloed in Computer Science Departments, but this is changing

“Software is eating the world”

Marc Andreessen



Figure 5.4: Whatever world you enter after you graduate, software has either eaten it, currently eating it or working out how to do so. Andreessen explains why software is eating the world and your future with it. Portrait of Marc Andreessen by JD Lasica on Wikimedia Commons w.wiki/3V48 adapted using the Wikipedia app

as we'll see in this chapter. It's not that everyone should jump ship to Computer Science, but that:

- Computing is too important to be left to Computer Scientists
- Computing is too important to be left to men (Spärck-Jones and Runciman, 2007)

Whatever subject you are currently studying, adding some computing to your education will empower you with the computational thinking skills you need to be an active producer, not just a passive consumer in modern society. Computing can open up new opportunities for you and improve your social mobility.

5.4 Computing is eating the world

Besides the social arguments, there are also strong economic reasons for studying computing. It's not just software that's eating the world, but its combination with hardware that dominates the list of the world's largest corporations by market capitalisation, shown in figure 5.5. What use is software without hardware?

Even if you don't want to work for any of these global oligopolies, their success is good news for *all* students of computing because it shows how important com-

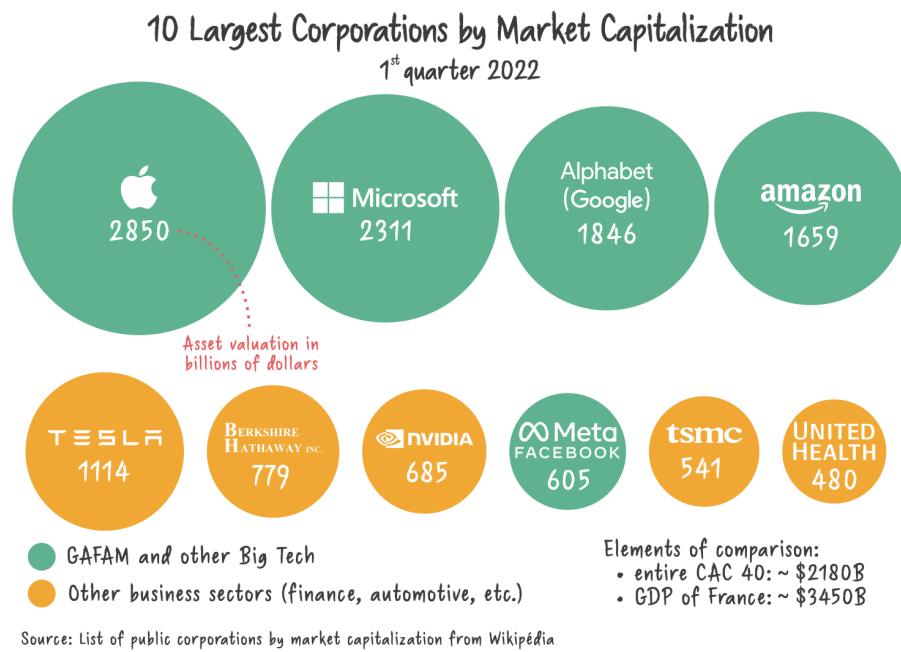


Figure 5.5: If stock markets are anything to go by, computing is eating the world. It would be impossible for Big Tech companies like Apple, Microsoft, Amazon, Alphabet (Google) and Meta (Facebook) to exist without computing. The ten largest corporations by market capitalisation (as of 2022) graphic by YBSLE/laboussole.coop on Wikimedia Commons w.wiki/3KEU

putation is to society, both commercially and otherwise. Another visualisation of data in figure 5.5 is shown in figure 5.6.

What figure 5.5 and figure 5.6 show is that computing is eating the stock market. This means that commercial demand for software developers is high, comparable to teaching and nursing in terms of raw numbers. In the UK, the most common jobs for graduates from 2019 are shown in figure 5.7, based on data taken from an update on the graduate labour market in 2022 (Ball, 2022b)

So if you can develop software, there's lots of choice and opportunities on offer. Although the data in figure 5.7 is from the UK, the story is the same in many other countries around the world.

Demand for developers is high, with many job vacancies going unfilled.

5.5 Passive consumer or active producer?

All this choice is a great thing but what sort of role do you want computing to play in your career? You can either be a passive consumer of computing or you can be an active producer, shaping the world of computing to get what you want from it, rather than what it wants from you. Going back to Andreessen's eating analogy in section 5.3, the choice is to be an *eater* or be *eaten*. Or to use a gaming analogy, *play* or be *played*.

5.6 Play your joker: Computational joker

Because of its social and economic importance, computing also gives you flexible career options. If academic disciplines are playing card suits, then Computer Science is the joker in the pack shown in figure 5.8. A versatile card, the computational joker can be played with (and without) *any* of the traditional four suits: diamonds, clubs, hearts and spades. That's because computing is a science *and* an art. It allows us to study human society and culture, so it's part of the humanities too (see digital humanities and computational social science for example). Last but not least, computing is also an engineering discipline and a branch of mathematics too. What all this means is that the computational joker is a wild card that can be played *whenever and wherever* you like, making it an incredibly powerful but dangerous card, depending on the game you are playing (see chapter 9).

The flexibility of computing as a career means you have a broad range of options on where you can apply your computational skills. You don't *have* to be studying Computer Science to take advantage of these opportunities, but it helps.

5.7 Summarising computing your future

Too long, didn't read (TL;DR)? Here's a summary:

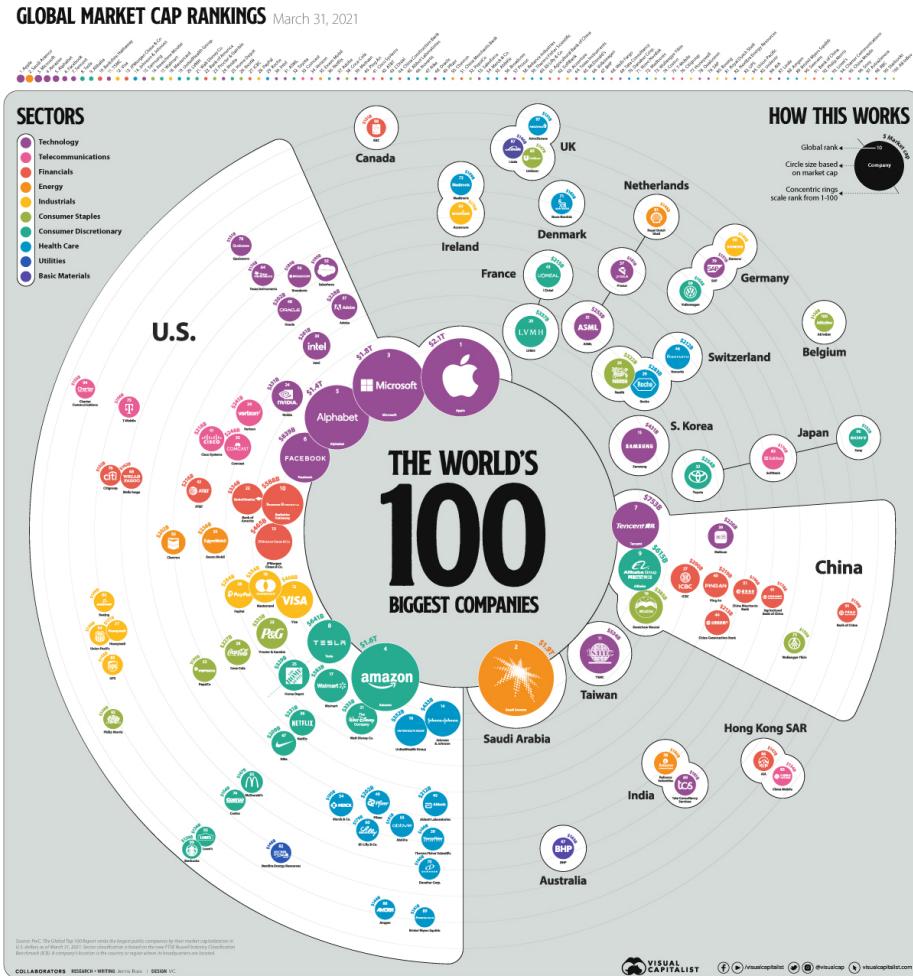


Figure 5.6: The Biggest Companies in the World based on market capitalisation data from PriceWaterhouseCoopers (PwC), as well as the countries and sectors they are from. Again, note the dominance of software and hardware: Apple, Microsoft, Alphabet (that's Google), Facebook and Amazon. Visualisation by the visualcapitalist.com at The Biggest Companies in the World in 2021 (Ross, 2021)

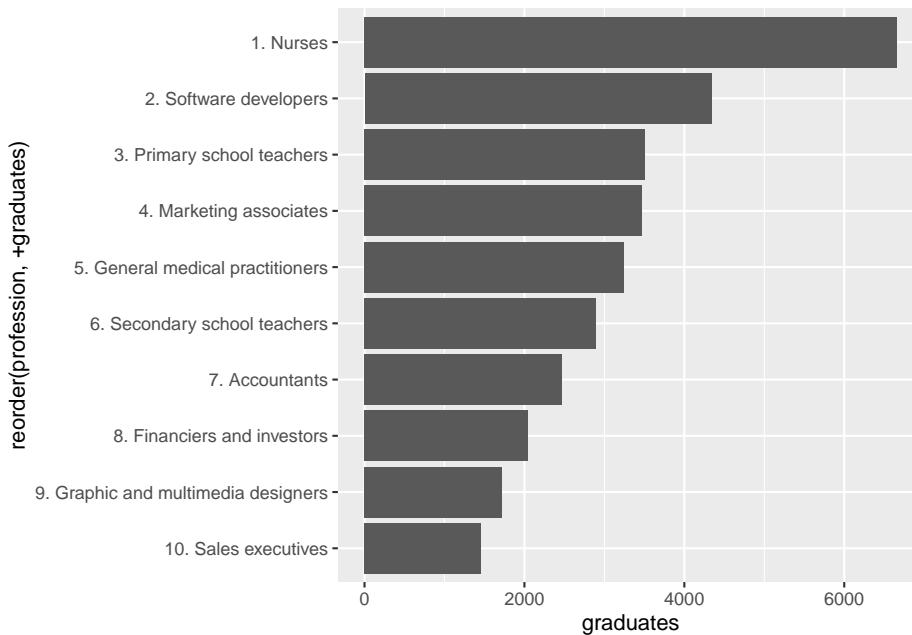


Figure 5.7: The top ten professions in the UK in 2019 based on numbers of graduates starting work, shown on the x axis. In 2019, 4340 graduates started a job in the UK job as a software developer (or programmer), with many developer vacancies unfilled. So, demand for software developers is high, comparable to teachers and nurses, according to data published by prospects.ac.uk (Ball, 2022b)

The Joker can be an extremely beneficial, or an extremely harmful, card. In Euchre it is often used to represent the highest trump. In poker, it is wild. However, in the children's game named Old Maid, a solitary Joker represents the Old Maid, a card that is to be avoided.

Joker (playing card)



Figure 5.8: If academic disciplines are playing card suits then Computer Science is the joker in the pack. Just like computing, the joker can be an extremely beneficial card, or an extremely harmful one. Public domain image of the Jolly Joker, a vintage Masenghini Italian playing card via Wikimedia Commons [w.wiki/35EW](https://commons.wikimedia.org/w/index.php?title=File:35EW&oldid=3535110) adapted from the joker playing card using the Wikipedia app.

There are lots of opportunities in computing but you don't *need* to have studied Computer Science to take advantage of them.

This chapter is under construction because I'm using agile book development methods, see figure 5.9.

The Death Star is a
fictional mobile space
station and galactic
superweapon featured in
the Star Wars space-
opera franchise

Death Star

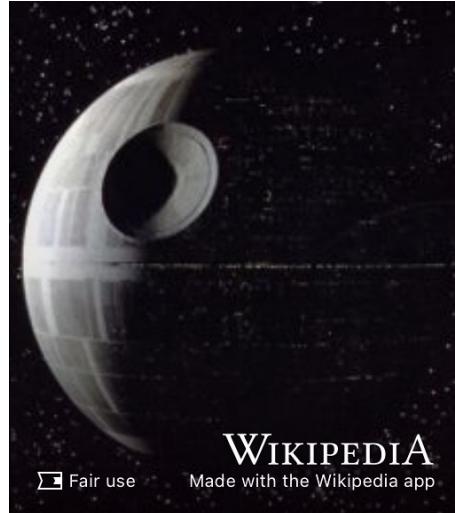


Figure 5.9: Just like the Death Star, this galactic superweapon book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Part II

TESTING & BUILDING

Chapter 6

Debugging your future

It's all very well *designing* your future but now you need to actually engineer it by *testing* and *building*. An obvious place to start is with your CV or résumé, because that's where most people get going. How can you create a bug-free CV, résumé or completed application form? How can you support applications with a strong personal statement or covering letter? These documents are a crucial part of your future so how can you debug them?

6.1 What you will learn

By the end of this chapter you will be able to

1. Structure and style the content of your CV and résumé appropriately
2. Articulate your story¹ as evidenced by your experience, projects and education etc
3. Identify and fix bugs in CV's by:
 - Their content (what you've done and who you are) and presentation (how you style your CV)
 - Constructively criticising other people's CVs
 - Asking for, listening to, and acting on constructive criticism of your own CV
4. Quantify and provide convincing evidence for any claims you make you on your CV
5. Identify gaps (mostly content bugs) in your own CV and how you might begin to fill them

¹actually a collection of short stories

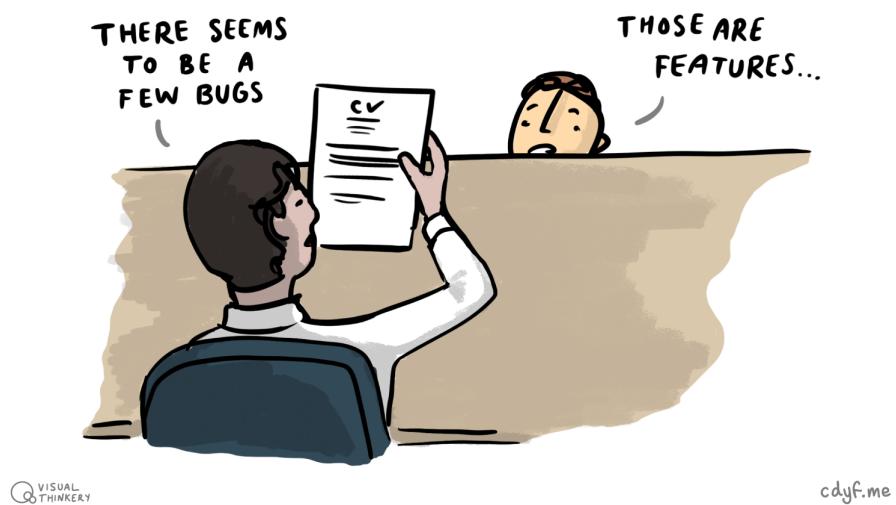


Figure 6.1: Is that a bug or a feature in your CV? To stand a chance of being invited to interview, you'll need to identify and fix any bugs in your written applications. If you don't, your application risks being sucked into a black hole and will never be seen again. Features not bugs picture by Visual Thinkery is licensed under CC-BY-ND

6.2 Beware of the black hole

Before we get started, let's consider some advice from software engineer Gayle Laakmann McDowell. Gayle is an experienced software engineer who has worked at some of the biggest technology employers in the world: Apple, Microsoft and Google. She has also authored a cracking series of books on technology careers, particularly *Cracking the Coding Interview* (McDowell, 2015) which we'll discuss in chapter 10. Gayle refers to the employer "black hole" described in figure 6.2.

The online application system – or, as it's more appropriately nicknamed, "The Black Hole," – is littered with so many resumes that even a top candidate would struggle to stand out.

Gayle Laakmann McDowell

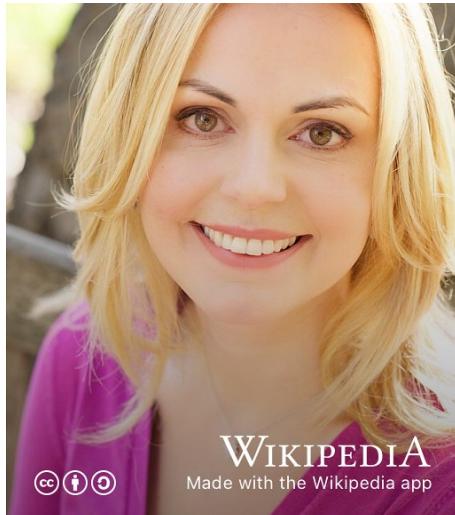


Figure 6.2: Beware of what software engineer Gayle Laakmaan McDowell calls the employer "Black Hole", especially if you're applying to large employers. "Getting through the doors, unfortunately, seems insurmountable. Hoards of candidates submit résumés each year, with only a small fraction getting an interview. The online application system – or, as it's more appropriately nicknamed, *The Black Hole*, – is littered with so many résumés that even a top candidate would struggle to stand out." (McDowell, 2011, 2014) Laakmann portrait by Gayle Laakmaan is licensed CC BY 4.0 via Wikimedia Commons w.wiki/wiu adapted using the Wikipedia app Thank you Gayle for permission to use your photo.

If you're applying to big employers, you'll need to create a CV that is good enough to stand out before it disappears over the event horizon and into an employers black hole, see section 8.4.8. Employer practices vary, but according to some estimates the person reading your CV will typically:

- Spend around **7 seconds** (or less!) deciding if it is interesting
- Shortlist about **20%** of the all CVs they get sent
- Interview about **2%** of the candidates on their shortlist

So, the FIRST thing your CV or résumé needs to do is, *persuade an employer*

to invite you to an interview. You can start with an employer-agnostic CV but you may need to come back and revisit the issues in this chapter once you have identified some target employers, so that you can customise and tailor your CV and written applications.

6.3 The CV is dead, long live the CV!

Résumés and CV's have reigned supreme in the kingdom of employment and recruitment for many years but their demise has often been predicted, see figure 6.3.

"The CV is Dead. Long
Live the CV!"

The king is dead, long live
the king!

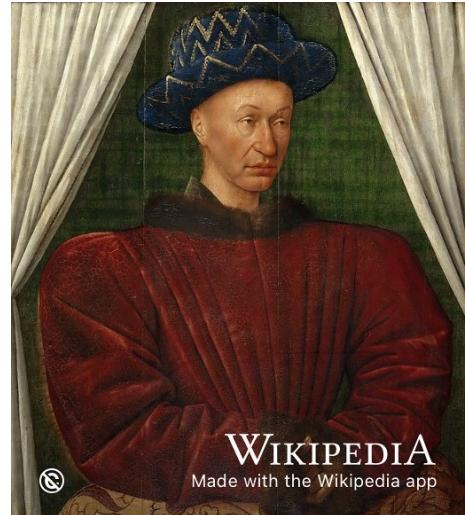


Figure 6.3: The seemingly contradictory phrase the king is dead, long live the king simultaneously announces the death of the previous monarch and the accession of a new one. Likewise, the death of the CV (and résumé) has long been hailed but it keeps coming back to reign again. The CV is dead, long live the CV! The résumé is dead, long live the résumé! Public domain image of a painting of Charles VII of France by Jean Fouquet on Wikimedia Commons w.wiki/3e3K adapted using the Wikipedia app

While it is true that some employers don't accept CVs or favour online application forms or digital profiles, it is still worth writing a CV because:

- Your CV provides a record for you of relevant things you've done
- Your CV enables you to get your story straight, *What's your story, coding glory?* see section 2.2.2 (Gallagher, 1995)
- Your CV forces you to articulate who you are and what you want, see chapter 2
- Some employers will still ask you for one, see section 6.7.6 and figure 6.3

- Your CV helps you script and practice your “lines” for interviews, see chapter 10

Even if an employer never asks you for your CV, you’ll frequently want to use the *content* of your CV in applications and interviews. So this chapter looks at how you can debug your written applications, whatever the mechanism is for submitting them.

You can of course *augment* your CV with various public digital profiles such as LinkedIn, Github etc. A public digital profile can help employers find you, see section 8.3.3.

Wherever you’re putting information about yourself for potential employers, on a CV, Résumé, LinkedIn profile the same principles apply. See the checklist in section 6.9 and for specific advice on LinkedIn see:

- [linkedin.com/learning/learning-linkedin-for-students](https://www.linkedin.com/learning/learning-linkedin-for-students)

6.4 It's not a bug, it's a feature!

An age old trope in Computer Science that engineers use to pass off their mistakes as deliberate features is the bug/feature distinction, see 6.4.

**“It's not a bug, it's a
feature”**

Software bug



Figure 6.4: “It’s not a *bug*, it’s a *feature*...” (according to A. Hacker). Do you have software bugs or undocumented features on your CV or résumé? Although tolerated in software, bugs in your CV, résumé and written applications can be fatal. Picture of gold-dust weevil *Hypomeces pulviger* by Basile Morin is licensed CC BY SA via Wikimedia Commons w.wiki/3E62 adapted using the Wikipedia app

Nobody likes buggy software, but unfortunately we routinely have to tolerate badly-designed, low quality, bug-ridden software in our everyday lives. (Mann, 2002; Charette, 2005)

In contrast, buggy CVs are rarely tolerated, they will usually end up in the bin. Even a tiny defect, like an innocent typo, can be *fatal* fatal. Most employers (particularly large and well known ones) have to triage hundreds or even thousands of CV's for any given vacancy. This means the person reading your CV is as likely to be looking for reasons to **REJECT** your CV, than **ACCEPT** it, because that's a sensible strategy for shortlisting from a huge pool of candidates. A buggy CV, application and covering letter could ruin your chances of being invited to an interview, see chapter 10.

Like writing software, the challenging part of writing a CV isn't the *creation* but in the *debugging* and subsequent re-writing. Can you identify and fix the bugs before they are fatal?

Coding Caution

If you ask three people what they think of your CV, you will get three different and probably contradictory opinions. CV's are very subjective and very personal. The advice in this chapter is based on common sense, experience and ongoing conversations with employers. What makes a good CV will depend on the personal preferences and prejudices of your readers. So, this chapter just describes some general debugging guidelines, rather than rigid rules.

It is good practice to *triangulate* responses from your readers. For example, if three different people point out the same bug in your CV you can be confident that it is a bug that needs fixing.

While referring to this guide, remember that:

- The main purpose of your CV is to get an interview, not a job. Your CV should catch attention and provide talking points for an interview
- Your CV will be assessed in seconds, rather than minutes so brevity really is key
- Bullet points with verbs first (see section 6.7.4) will:
 - allow your reader to quickly skim read your CV because employers don't read CVs quickly (Richards, 2019b)
 - highlight your key activities
 - avoid long sections of flowery prose which the reader will very probably skip anyway. Instead, you can use your beautiful prose on your covering letters or personal statements, see section 6.10

You're not trying to tell your *whole* life story from section 2.2.2 but to distill the essentials into several short stories which can be summarised into a handful of bullets or sentences. It's a bit like the blurb or synopsis on the back of a novel, can you entice the reader into wanting to find out more (by inviting you to an interview)?

6.5 Is it a bug or a feature?

Wherever criticism of your CV comes from, don't take it personally - it is probably one of the first you have written. Think of your current CV as an alpha or beta version that you continuously test, release and redeploy. There are many chances to debug and improve your CV during your study but before potential employers read it. The aim of this chapter is to help you improve your CV, whatever stage you are at. Employers often grumble that Computer Science graduates lack written communication skills. Written applications and CV's are a common example of this.

1. **EDUCATION:** Is your year of graduation, degree program, University and expected (or achieved) degree classification clear? Have you mentioned things you are studying now, not just courses you have finished? See section 6.6.2
2. **STRUCTURE:** Is the structure sensible? Is it in reverse chronological order? Most important (usually recent) things first? Not too many sections or anything missing? See section 6.6
3. **NUMBERS:** Have you *quantified* the context, action, results and evidence of the short stories on your CV where possible? See (C.A.R.E.) in section 6.7.2
4. **VERBS FIRST:** Have you talked about what you have actually done using prominent **verbs**, rather than just what you think you know? Avoid long sections of prose, see section 6.7.4
5. **STYLE:** Does it look good, with a decent layout with appropriate use of LaTeX or Word or whatever? (Hull, 2022) Are there any spelling mistakes, typos and grammar? Don't just rely on a spellchecker, some typos can only be spittted spotted by a human reader see section 6.7.7
6. **LENGTH:** Does it fit comfortably on (ideally) one page (for a Résumé) or two pages (for a CV)? See section 6.7.3
7. **SEE ALSO:** This is just a quick checklist to get you started, see the longer CV checklist in section 6.9

6.6 Structure your CV

How you structure your CV will depend on who you are and what your story is. Recruiters at Google suggest you have four or five sections, that follow a header section. Before we look at those, lets look at some general points about CVs, watch the video shown in figure 6.5.

As Jonathan Black, director of the careers service at the University of Oxford has pointed out, (Black, 2019b) there are three key things you want to demonstrate in your CV:

1. that you can take responsibility
2. that you achieve things
3. that you are good company (or at least a valued team member)

Figure 6.5: Recruiters at Google, Jeremy Ong and Lizi Lopez outline some tips and advice for creating your résumé. You can also watch the 8 minute video embedded in this figure at youtu.be/BYUy1yvjHxE. (Ong and Lopez, 2019)

How can you demonstrate each of these traits? Watch the short video in figure 6.6.

Figure 6.6: Jonathan Black, head of the careers service at the University of Oxford, explains how to create a top notch CV by replacing meaningless assertions with meaningful evidence. You can also watch the 11 minute video embedded in this figure at youtu.be/yjdvCHWVtE4. (Black, 2019b)

Quantifying and providing evidence for any claims you make will create a much more convincing CV. Numbers can transform meaningless assertions described in figure 6.6 into meaningful evidence. So for example:

* **Achieved excellent results**

...is a bit vague, what were the results exactly, can you measure them somehow? Or at least describe them?

* **Worked in a team**

So you worked in a team? **Worked** is vague. What was your role and contribution exactly? How long did the project last? How many people were on your team? What were the results of your actions? What's the story, coding glory? (Gallagher, 1995)

6.6.1 Your header

The first thing in your CV is the header, a simple section giving your name, email, phone number and any links shown in the CV in figure 6.7 for Alan Turing. That's it!



Figure 6.7: Keep the header of your CV simple like Alan's example here. Just your name, email, phone number and any relevant links (e.g. enwp.org/Alan_Turing here) are all you really need. If you have a public digital profile, your header is a good place to put it, see section 8.3.3. Any additional information risks wasting valuable space and distracting your reader. Less is more.

Your header doesn't need to include any more information than your name, email, phone and any links. This means your birth date, marital status, photo² and home address aren't relevant. You don't need to give multiple phone numbers or emails either, just one of each will do. If an employer wants to invite you to an interview, they'll get in touch by email, phone (or possibly LinkedIn, github etc) so other contact details such as your postal address are an irrelevant distraction at this point. After your header I suggest you have about five sections that cover some or all of the following:

1. **Education:** the formal stuff, see section 6.6.2
2. **Experience:** paid work, see section 6.6.3
3. **Projects:** personal, school, social, side or University projects see section 6.6.4
4. **Leadership and awards:** see section 6.6.5
5. **Optional:** but don't call it that, see section 6.6.6

Let's look at each of these sections in turn:

6.6.2 Your education

Unless you have a *significant* amount of experience, the **Education** section of your CV should be the first real section, after the header see figure 6.8. Your education section needs to strike a balance between:

²different countries have different conventions when it comes to passport-style photos on a CV. In the UK what you look like should *not* be a factor in an employers decision to interview you, so leave it out

- Describing in enough detail what you've studied and any projects you've completed at University as part of your formal education
- Keeping it short and sweet by avoiding getting bogged down in tedious and irrelevant details will be of little interest to your readers

At the 1996 Labour Party conference, Blair stated that his three top priorities on coming to office were "education, education, and education"

Tony Blair



Figure 6.8: Tony Blair's top three priorities as a new Prime Minister of the United Kingdom were **education, education, education**. (Coughlan, 2007) Likewise, your education should be a top priority on your CV, it should go first, unless you've got a *significant* amount of **experience**. As a graduate (or undergraduate) your education is the most recent and relevant thing about you so it should have top billing, see the discussion in figure 6.5. This GNU Free Documentation Licensed (GFDL) portrait of Tony Blair in 2007 by the Polish Presidency on Wikimedia Commons w.wiki/64Kd has been adapted using the Wikipedia app

You've invested a significant amount of time and money in getting your degree. At this stage, your degree justifies a longer description than a rather terse one line **BSc Computer Science**. You'll need to:

- say *more* than Pen Tester and Rick Urshion, who haven't said enough about their University education
- say *less* than Mike Rokernel, who has given *way* too much information on his degree.

You don't need to name every single module, course code with a mark for each. Neither do you need to give your result to FOUR or FIVE significant figures: 71.587%³. Two significant figures will do just fine: 72% (**first class**). You might like to pick out relevant modules, or the ones you enjoyed most. Employ-

³Do you *really* think academic performance can be measured that precisely?

ers like Google encourage applicants to emphasise courses on data structures and algorithms, but you'll need to tailor your description to the role and be brief. On a one page CV, you might only have two or three lines to describe your higher education.

6.6.3 Your experience

Experience is where you can talk about any paid or voluntary work experience you have. For paid work call it **EXPERIENCE** rather than **WORK EXPERIENCE** as the latter can imply work *shadowing*, see section 7.3. Work shadowing is valuable, but paid work is even better so you should make it clear if your experience was paid or not. Don't discount the value of casual labour, such as working in retail or hospitality, these demonstrate your work ethic and ability to deal with customers, often under pressure. You are more than just a techie too, so anywhere you've worked in a team is experience worth mentioning, even if that team was just two people. Two people is still a team.



Figure 6.9: Are you experienced? What have you done outside of your formal academic education? Employers will want to know *much more* about you than just your **Education**. Are you **Experienced**? What **Projects** have you done? Can you demonstrate **Leadership**? Have you won any **Awards**? Experience sketch by Visual Thinkery is licensed under CC-BY-ND With apologies to Jimi Hendrix. (Hendrix, 1967)

If you don't have much experience, don't worry, there are plenty of opportunities to get some more . For details, see chapter 7 *experiencing your future*.

Are you experienced? Whatever it is, make sure you add it to your **experience** section.

6.6.4 Your projects

The PROJECTS section of your CV is where you can describe all other activities you get up to. These might include:

- personal side projects, see section 7.3.6
- social responsibility projects, see section 7.3.5
- open source projects, see section 7.3.4
- entrepreneurial projects, your “side hustles”
- hackathons, olympiads and other competitive or collaborative events, see section 7.3.6
- University projects
 - these sometimes fit better in your Education section
 - Be *careful* publishing coursework, this can make you liable for accusations of plagiarism. If someone copies your coursework, you will **both** be guilty of academic malpractice...
 - ... but final year (“capstone”) projects are great as these will tend to be unique, distinguishing you from everybody else in your year group

Your projects will most likely be unpaid because paid work tends to fit better under the heading **experience**, see section 6.6.3. Perhaps you’ve completed some courses outside of your education such as a massive open online course (MOOC) or similar. Hackathons and competitions, fit well here too. (Fogarty, 2015) You don’t *need* to have won any prizes or awards, although be sure to mention them if you have. Participating in hackathons and competitions clearly shows the reader that you enjoy learning new things. Demonstrating an appetite for new knowledge and skills will make your application stand out. If you’re looking for some inspiration for side projects, the codecrafters-io/build-your-own-x repository is a good starting point. Building and creating new things is a great way to understand them, just ask Richard Feynman shown in figure 6.10. What you cannot create you do not understand. Another great way to learn is to get involved with open source projects which we describe in section 7.3.4.

Any longer projects you’ve done at University are worth mentioning. Your projects are important because they differentiate you from everyone else in your year group. Try to be *more* descriptive than this:

* First year team project

or perhaps

* Second year team project

or even just

* Final year project

By themselves, those project names are pretty opaque. They are OK for giving the context of your story but don’t give the reader much else to go on. What was the story (the context, action, result and evidence (CARE) we described in

**“What I cannot create I
do not understand”**

Richard Feynman

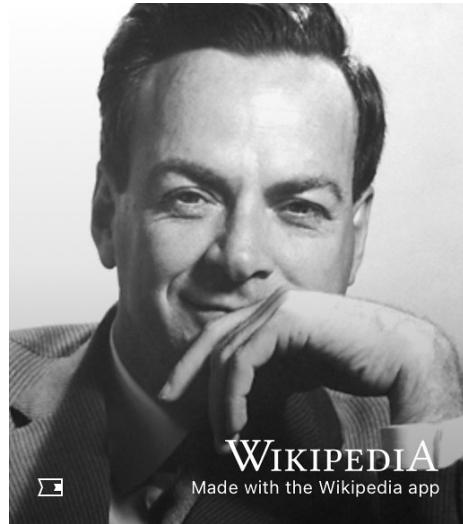


Figure 6.10: Physicist Richard Feynman once chalked “What I cannot create, I do not understand” on his blackboard at the California Institute of Technology where he taught The Feynman Lectures on Physics. (Feynman, 1988) Creating software and hardware in personal side projects is a great way to build new understanding *and* help your CV stand out see github.com/coderaiders-io/build-your-own-x. Public domain image of Richard Feynman by The Nobel Foundation on Wikimedia Commons w.wiki/3Xoy adapted using the Wikipedia app

section 6.7.2) of those projects? How many people were in your team? How long did you collaborate for? What did you build? What was it called? What did it do? What roles and responsibilities did you have in the team? Was their conflict in the team? How did you resolve it? How did you motivate the free-riders in the team to contribute? This is all excellent CV fodder, see 6.11

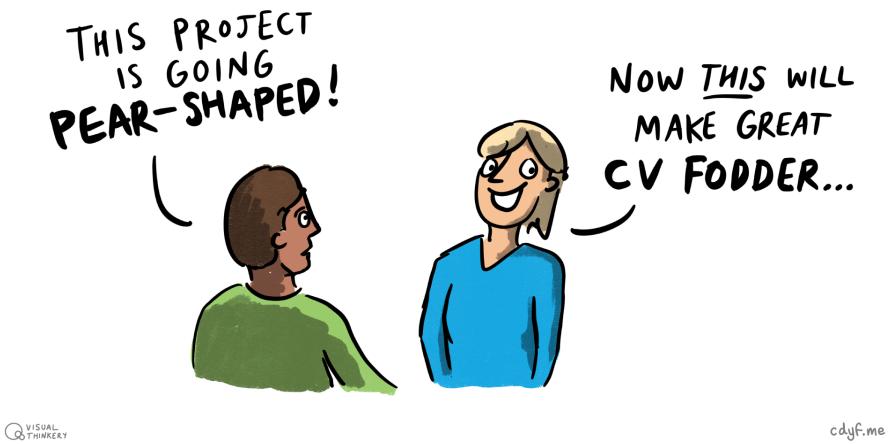


Figure 6.11: Have you ever had a project that didn't go very well? Did it all go pear-shaped? Perhaps you managed to turn the project around? Maybe you learned some lessons from those painful mistakes you won't be repeating? Difficult projects can make great stories (aka CV fodder) because mistakes *can* be good, see section 14.5, as long as you don't repeat them. CV fodder sketch by Visual Thinkery is licensed under CC-BY-ND

It's often better to describe what **YOU** did before you describe what the software, hardware or project did. Your reader is much more likely to be interested in the former than the latter. Let's imagine you've developed a piece of software called `WidgetWasher`. You might describe it like this:

- * `WidgetWasher` is a web service that washes widgets
- * It uses an HTTP API and secret keys
- * Tested `WidgetWasher` on a range of different operating systems
- * Collaborated with one other contributor over two days
- * Designed and implemented an API

Instead, you could reverse ↑↓ the order to change the emphasis like this:

- * Designed and implemented an API ↑
- * Collaborated with one other contributor over two days ↑
- * Tested `WidgetWasher` on a range of different operating systems
- * Utilised an HTTP API and secret keys ↓

* Created a web service that washes widgets called WidgetWasher ↓

The latter has all the same information, but by reversing the order, you've emphasised what *you* did, rather than what the software did. Much more interesting.

6.6.5 Your leadership & awards

If you can demonstrate leadership, you may want to dedicate a whole separate section for it. If you have experience of tutoring or mentoring other students, perhaps younger or more junior ones, make sure you highlight it as leadership. Teaching is a different kind of leadership to running a small business or captaining a sports team, but it is leadership nonetheless.

A leadership section is also a good place to add any prizes you've won. If you've been granted any interesting awards or honours be sure to mention them, because they show the reader that a third party has endorsed you. *Other* people think you are good at something, and this says much more than *you* thinking you're good at something. You'll typically need a bit more than this:

* Awarded the Poppletton University scholarship for excellence

Congratulations, but how many people were awarded that prize? How many applicants or entrants were there and what percentage were successful? Was it a regional, national or global award? How frequently is the award given? It is unlikely that your reader will have heard of the award unless it is widely known. You might talk about rank e.g.

* Won third prize in the flibbertigibbet competition

Again, congratulations, but third place out of how many:

- 3/3? Not so impressive
- 3/30?
- 3/300?
- 3/3000? More impressive

Numbers make a big difference so if you're going to mention awards, quantify⁴ where you can so the reader can understand what the prize was for. You may well be intimately acquainted with the flibbertigibbet competition but your reader probably won't have heard of it, so give some context.

6.6.6 Your optional extras

If you have anything else you want to highlight besides your Education, Experience, Projects and Leadership and Awards you *may* still have room for one more optional extras section. Try to come up with a better name than

⁴an approximation will do if you don't have exact figures

Miscellaneous, Extracurricular or Other highlights and activities, which sound like dumping grounds for the leftovers. You might decide to have a dedicated skills section but see section 6.6.7.

You may opt to have a hobbies and interests section which can add a bit of colour to what can otherwise become quite a dull list of facts. However, it is debatable how many (if any) hobbies and interests you should list on your CV. For a one-pager you're usually pushed for space and looking for things to edit out (rather than add in), so if you're going to list hobbies, I'd stick to those that are relevant to the job or those that demonstrate particular transferrable or soft skills. Organising or participating in team sports is a good example of a relevant hobby as it provides evidence of your teamwork, commitment, reliability etc, see figure 6.12. Other collaborative activities outside of sport will also demonstrate your communication skills. (Cheary, 2021)



Figure 6.12: It is debatable which, if any, of your hobbies should go on your CV. You might get lucky and your interviewer(s) will share your esoteric passion for Quidditch (say). (Rowling, 1997) However, such coincidences are unlikely so if you *do* list your hobbies and interests, think carefully about *WHY* you are mentioning them and *WHAT* skills or knowledge you are trying to demonstrate to potential employers. Anyone for Quidditch sketch by Visual Thinkery is licensed under CC-BY-ND

The fact that you enjoy swimming, reading, football, cooking and hiking is vague and pretty boring, making it unlikely to be a factor in the decision to invite you to interview. There's nothing *wrong* with these pastimes but there's not much point mentioning them on your CV unless you have, for example:

- coached a swimming team
- trained as a mountain leader

- organised, hosted and participated in a book club
- won some awards for your hobby, like a black belt for example
- regularly participated in or organised competitive sports matches (for example)

In this case, the *way* you have engaged with your hobbies demonstrates your communication and leadership skills. So in this context, your hobbies *are* worth talking about if you have any space left. However, if they are just hobbies that enable you to amuse yourself, you should probably leave them out as they are unlikely to be of interest to an employer. Sure, they add colour to your CV, but you've probably got more interesting and important things to say about who you are.

Of course, you *might* get lucky and your interviewer(s) could be intrigued by (or even share) your esoteric passion for Quidditch (say), but you can't rely on it. So, I reckon if you're going to mention hobbies and interests at all then ...

- pick the unusual hobbies that make you unique
- describe the interests that add some colour and personality to your CV
- be specific, e.g. `reading japanese novels from 20th century` might sound more interesting (and convincing) than just `reading`
- highlight any `actions` you've taken, see chapter 18

... or just leave them out altogether. That option is up to you.

6.6.7 Your skills?

You may be tempted to dedicate a whole section on your CV to `skills`, particularly the technical ones. Maybe it makes you feel good listing them all in one place like a stamp collection. If you're going to have a `skills` section, keep it short. Why? Let's imagine, that like Rick Urshion, you include Python in a long list of skills, with its own dedicated section. There are at least five problems with Rick's not so skilful approach:

1. **No Context** to give the reader an idea of where he's developed or used his Python skills. Was it during his education, as a part of his work experience or his personal projects? We don't know because he doesn't say.
2. **No Actions** to demonstrate what he's *done* with his Python skills. So Rick claims he knows Python. So what? What did he use them for? We don't know because he hasn't told us.
3. **No Results** given for what the outcome of using the skills was. Did he save his employers some money? Did he make something more efficient? Did he learn some methodology? We will never know.
4. **No Evidence** to support his claims. Perhaps he DOES have Python skills, perhaps he DOESN'T. Is he telling lies and peddling fake news (see

section 8.4.4)? It's difficult to tell.

5. **No C.A.R.E.** There's no story told for that skill, see figure 6.13. This makes for a very dull and boring read. Yawn. NEXT!

(What's the Story)
**Morning Glory? was
inspired by Noel's friend
Melissa answering the
phone with said phrase**

(What's the Story)
Morning Glory?



WIKIPEDIA

Made with the Wikipedia app

Figure 6.13: (What's your Story) Morning Coding Glory? (Gallagher, 1995) What is the Context, the Actions, the Results and the Evidence for the stories that you are trying to tell? Show your C.A.R.E. in storytelling. CC BY portrait of Noel Gallagher by alterna2.com on Wikimedia Commons w.wiki/3bimy adapted using the Wikipedia app

So, this doesn't mean Rick shouldn't mention his Python skills. Where he can, he needs to give us the Context, Action, Result and Evidence (C.A.R.E.) of his story described in section 6.7.2. This will make his Python story much more convincing and interesting to read. Showing a bit of C.A.R.E. will improve his chances of being invited to interview.

This applies to soft skills too, not just hard technical skills. It is often better to mention the context in which you've used the skills you mention on your CV. So, if you're going to have a skill section:

- keep it short (one or two lines maybe) but personally I'd avoid dedicating a whole section to it
- stick to your strongest and most relevant skills that you are comfortable to answer questions on in your interview, rather than an exhaustive encyclopaedic inventory
- avoid listing mass marketed office products of Microsoft (e.g. Word etc) as a skill, they are *not* generally very interesting skill because almost *everyone* has them. They won't set you apart much from your competition, so don't waste valuable space talking about office unless you've done something

interesting with them, like some advanced integration with other software. Cloud services can be a slightly different matter, see section 11.4.1.

If you're a computer scientist, you also have demonstrable **meta** skills like the ability to learn things quickly. You can also think logically and critically, reason, problem solve, analyse, generalise, decompose and abstract - often to tight deadlines. These computational thinking skills are future-proof and will last longer than whatever technology happens to be fashionable right now. Employers are often more interested in these **meta** capabilities and your potential than in any specific technical skills you may or may not have.

6.7 Birds eye view

Having looked at the sections you're likely to have, we'll take a birds eye view of your whole CV. The issues in this section apply to the whole of your CV, rather than individual sections.

6.7.1 Your style

Making your CV look good can take ages, but a well presented CV will stand out. While it's worth making an effort to style carefully and consistently, you need to be wary of the huge time sink of typography.

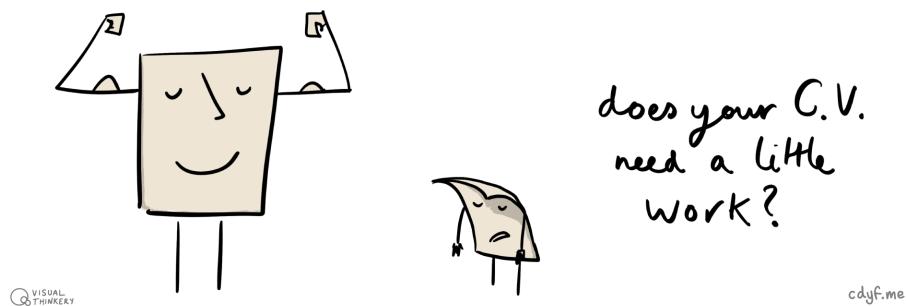


Figure 6.14: Does your CV need a little work? The truth is your CV is never finished, you will be continuously developing, debugging and releasing it throughout your life. It's such a crucial document because it will determine if you are invited to interview. Any time you spend getting it right is a good investment. CV work sketch by Visual Thinkery is licensed under CC-BY-ND

Seemingly trivial typographical details can make a big difference to the overall look and feel of your document. The devil is in the detail, for example this text in a bulleted list doesn't use a proper indent...

- The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy

dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog

... whereas the text below does use a proper indent. The latter looks more professional and is easier to read, especially when you have multiple bullet points:

- The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.
- The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

Whatever your typographical style is, portable document format (*.pdf) is the safest way to deliver it. It's called portable for a reason. While Microsoft Word is fine for editing, it is difficult to ensure that a Word document doesn't get mangled by transmission via the web, email or even Microsoft's own products like Teams! PDF is much safer, you can be more confident that it will work well on a range of different operating systems and devices. Try opening a Word document on a range of different smartphones and tablets and you'll see what I mean. It helps your reader if you can give your CV file a descriptive name, especially if it is being read by a human and not a robot.

- cv.pdf (not a very descriptive filename)
- ada_lovelace.pdf(a descriptive filname)
- cv-version1-final-FINAL-version-USE-THIS-ONE.pdf (reveals more about you than you might want to!)
- alan_turing.pdf(a descriptive filname)
- one-page-version.pdf (unhelpful)

It's fine to author your CV in Microsoft Word, but you'll want to save it as (or print to) PDF to make it more platform independent. LaTeX and overleaf can be used to create professional PDFs and there are plenty of good templates. See Getting started with LaTeX (LaTeX4year1) if you've not used LaTeX before, or you need to refresh your memory. (Hull, 2022)

6.7.2 What's your story, coding glory?

One way to structure descriptions of items within each section of your CV is to use **Context, Action, Result and Evidence** (C.A.R.E.) to tell your stories. What's your story, coding glory? (Gallagher, 1995) The C.A.R.E. method can also be useful for structuring answers to interview questions, especially if you get nervous. So for example, rather than just listing Python as a skill, you could tell the reader more about the context in which you've used python, what you actually did with it and what the result was. You really need to spell it out.

- **CONTEXT:** So you've used Python, but in what context? As part of your education? For a personal project? As a volunteer? In a competition?

All of the above?

- **ACTION:** What did you *do* with Python? Did you use some particular library? Did you integrate or model something? What verbs can you use to describe these actions, see chapter 18
- **RESULT:** What was the result and how can you measure it? You picked up some new skills? What was the impact? Perhaps you made something that was inefficient and awkward into something better, cheaper or faster? Some things are hard to measure but you should quantify results where you can.
- **EVIDENCE:** Where evidence exists, you should highlight it. That could be a quantification, for example describing a result in numbers (see *as measured by* below) or it might be certification described in chapter 12. If you're talking about software, point to a copy online if you can, see section 8.3.3. Nothing says "I can build software ..." quite like "... and here's one I made earlier".

You don't have to stick rigidly to the order C.A.R.E. as long as they appear somewhere. For example, recruiters at Google (see figure 6.5) advise candidates to describe their experience and projects using this simple pattern:

* Accomplished [X], as measured by [Y], by doing [Z]

Where *accomplished* is Result, *measured by* is Evidence and *doing* is the Action. So instead of just saying:

* Generated reports for end users

You could say:

* Generated daily reconciliation report for team by automating workflow of 8 different

The latter is better because it is more specific, captures the result (*accomplishment*), by giving evidence (*8 different tasks*) and talks about the actions (the *doing* part). Choose the verbs you use carefully, see chapter 18 for examples.

6.7.3 Your length

How long should your CV be? Many people start with a two page CV, which is a sensible starting point shown in figure 6.15. It is also advisable to create a one page Résumé. (David, 2017)

At this stage in your career you *should* be able to fit everything on to one page. However, it can be challenging and time consuming squeezing it all on, see figure 6.16.

It takes more time to write less. Writing a one page résumé is a valuable exercise, because it forces you to distill and edit out any filler or fluff, which you sometimes find on two page undergraduate or graduate CVs. It is much better

Figure 6.15: How long should your CV be? Should you write a two page European style CV or an American style résumé (one pager)? You can also watch the 5 minute video embedded in this figure at youtu.be/0abDOKHS5T0. (Hull, 2021a)

"I would have written a shorter letter, but I did not have the time."

Lettres provinciales

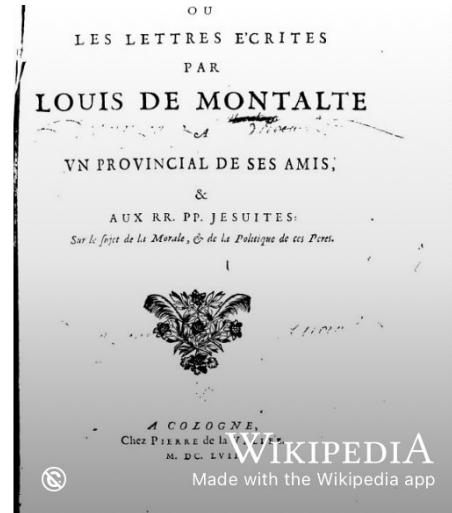


Figure 6.16: I would have written a shorter letter, CV, Résumé but I did not have the time. This quote (or meme) is frequently attributed to Blaise Pascal's *Lettres provinciales* (O'Toole, 2012). Public domain image by Gallica on Wikimedia Commons w.wiki/3Uzn adapted using the Wikipedia app

to have a strong one-page résumé than a weaker two-page CV that is padded out with filler to make up the space, as described in the video in figure 6.15. Adding more features (pages and content) to your CV doesn't necessarily make it better. Sometimes adding more features to your CV will make it worse, as shown in figure 6.17.

If you're struggling to fit all the information onto a one page résumé, revisit each section and item carefully. Is there anything you can drop? Can you save a wasteful word here, or a lazy line there? Check for any spurious line breaks because every pixel counts. Don't throw your two page CV away, it is still a good store of stuff you might want to add to customised one-page résumés.

6.7.4 Verbs first: lead with your actions

A simple but effective technique for emphasising what you have done, rather than just what you know, is to start the description of it with a verb. Employers don't just want to know what you know, but what you have actually done. So, for example, instead of saying e.g.

* In my second year CS29328 software engineering module I used Java, Eclipse and JUnit

you could say:

* Built and tested a large open-source codebase using Eclipse, Ant, JUnit and Jenkins"

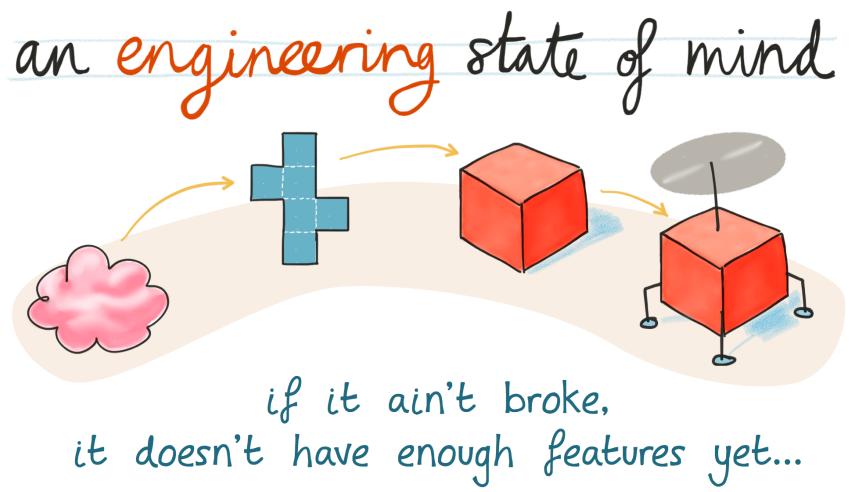


Figure 6.17: If it ain't broke it doesn't have enough features yet. Adding more features to software doesn't necessarily make it better. Likewise, adding more pages and content to your CV or résumé won't always improve it. It's often better to be precise and concise, rather than bloated and potentially more buggy. An engineering state of mind by Visual Thinkery is licensed under CC-BY-ND, with help from Dilbert cartoonist Scott Adams

followed by:

* Added and deployed new features to a Massively Multiplayer Online Role-Playing Game

The latter examples get to the point much quicker and avoid the problem of using I, me, my... too much which can sound self-centred and egotistical. Although your CV is all about you so it is natural to have a few personal pronouns in there, too many can look clumsy and give the wrong impression. Choose the verbs you use carefully, see chapter 18 for examples.

6.7.5 Your links

Augmenting your CV with web links (hyperlinks) can add important context to your story, without adding too many words or taking up valuable space. An example using LinkedIn is shown in figure 6.18



Figure 6.18: Adding links is a good way to augment your CV. If you're adding LinkedIn, make sure you customise your public profile URL, (the .../in/handle) to remove the default randomly generated alphanumeric string at the end, like the 038b37 example here. (Hoffman, 2020) You can also remove any ugly http, colons ::, forward slashes //, www and trailing / in URLs which are distracting noise. Just make sure links are clickable in the pdf, don't 404 if they are followed and work when printed on paper too. Neither do you need to waste valuable space telling people what the link is, like in the first example, the domain name already tells you that it is a LinkedIn profile.

Links can provide **Context**, (**Action**), **Result** and **Evidence** (see C.A.R.E. in section 6.7.2) by quantifying and substantiating any claims that you make. Links demonstrate what you've done and which communities you've been a part of. For example, you might say things like:

* Built a thing called example.com

This shows that the thing you built was substantial enough to have its own domain. It also says "I like building things. Look at this thing I built just for fun, it's really cool". Or you might say:

* Elected as a representative for hacksoc.com

The link shows you were part of a community: "I was part of a bigger thing..." you

might not have heard about but you can find out about here. You might also say:

* Competed at hack-to-the-future.com part III

Reading between the lines: “I really enjoy learning from other people by going to hackathons and competitions”

... and so on. So links are crucial features of your CV and an interested reader *may* even follow them. Treat links with respect and they will support your goals and help your readers. Invest some time thinking about how you word the link text, and how they would be understood out of context. Make sure that:

- your hyperlinks are readable and descriptive (Richards, 2019a)
- your hyperlinks are clickable in the PDF. Don’t expect your reader to cut and paste (or even type) URLs, they are too busy. If they are clickable, people are much more likely to follow them
- your hyperlinks are paper-proof. Some people still print CVs so links with opaque titles like “My LinkedIn”, click here, and “Projects on Github” won’t work well on printed paper unless you explicitly spell out the link. See to print or not to print — a CV, that is (Garone, 2014)

Besides LinkedIn you could include public profiles from github.com, devpost.com, hackerrank.com and stackoverflow, see section 8.3.3 (Hamedy, 2019). You can also link to personal projects, any articles you’ve published or your blog if you have one. Obviously, you need to be careful about what you link to and what employers can find out about you online. They *will* Google you. So keep it professional and, as we discussed in a section 3.8, be wary of social media.

Coding Comment

LinkedIn is much more than a tool for publishing your CV or résumé because it also allows you to:

- search and apply for jobs
- create a digital portfolio that is publicly available⁵
- connect with professionals, think social media for jobs

Chapter 8 looks at job searching in more detail but for now just note the similarities and differences between LinkedIn, CVs and résumés outlined in table 6.1. Some of these differences also apply to other places online where you might augment your CV with extra information such as github, devpost and others mentioned in section 6.7.5 and 8.3.3.

So, LinkedIn can be a useful tool for job hunting but it’s a different beast to traditional CV’s, some of these differences are shown in table 6.1

⁵you might want to be careful about what information you put out there!

Table 6.1: Comparison of LinkedIn with conventional CV's and résumés. The two mediums have a lot in common but also provide different communication channels which can be complementary, e.g. you include your LinkedIn profile on your CV to augment it.

LinkedIn	CV or résumé
Dynamic document that you can constantly update	Static document
Generally speaking, no length limit	Limited length
Public or semi-public document, more like social media	Private document
Only one profile, you can't easily transfer connections if you open more than one account	You might need multiple accounts
Generic, allows employers and recruiters to target you	Specific, controlled targeting

6.7.6 Robot proofing your CV

It's a good idea get feedback from as many different sources as you can on your CV. By *sources* I don't just mean humans, but also robots. Larger employers will use automated Application Tracking Systems (ATS) to log and trace your application. These "résumé robots" (if you like) are unlikely to have arms and legs like the one in figure 6.19, but they *will* be looking for keywords and standard headings in your CV. (Schellmann, 2022) You can get automated feedback from a range of different automated systems, though it is a good idea to remove any personal information like phone numbers and emails before using these free services. You might also want to check what the services privacy policy says about what they do with your personal data. Résumé robots include:

- careerset.io, a free service provided by a UK based company, careerset Ltd. If you're a manchester student login at careerset.io/manchester
- jobscan.co, a free service provided by an American company not really a CV checker
- google.com/search?q=resume+checker
- google.com/search?q=CV+checker

Besides providing feedback on the content of your CV, using these systems can help address issues such as the use of tables or layout which may cause problems for some systems. For example, some systems ignore the second column of a two-column CV because they can't parse the PostScript reliably to identify columns. Some things to check with automated CV screens:

- Have you used standard headings for the sections? Non-standard sections maybe ignored or misunderstood
- Have you used appropriate verbs to describe your actions?
- Is your layout and design robot friendly? Sometimes tables and two column layouts can get horribly mangled, see what happens to tables and columns in an applicant tracking system (Shields, 2019)

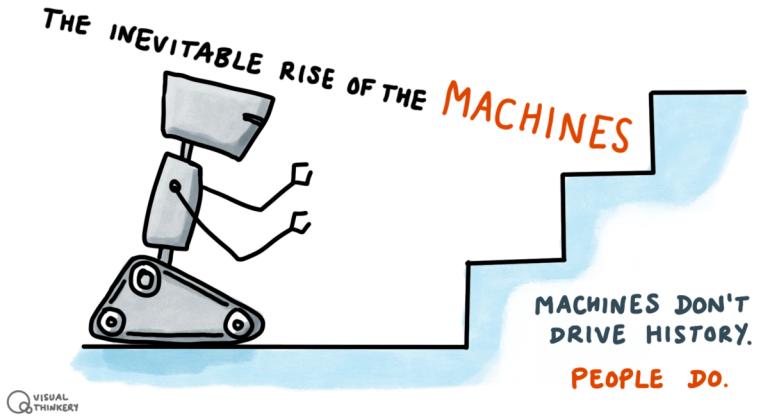


Figure 6.19: Just like the daleks in Doctor Who, résumé robots often struggle to get up the stairs. Although they typically don't have arms, legs (or even wheels) robots are likely to play an important role in decisions made about if you are worth interviewing, especially if you're applying to bigger companies. Make sure your CV is résumé robot friendly by getting feedback from a robot. Machines by Visual Thinkery is licensed under CC-BY-ND

6.7.7 Spellchecking

Bots, grammar and spell checkers will improve your CV but you can't rely on automated help completely. The résumé robots described in section 6.7.6 will just encode the biases and prejudices of whoever wrote their algorithms. Spellcheckers can't be relied on completely either, as shown in the poem below:

Eye halve a spelling checker
 It came with my pea sea
 It plainly marques four my revue
 Miss steaks eye kin knot sea

Eye strike a key and type a word
 And weight for it to say
 Weather eye am wrong oar write
 It shows me strait a weigh

As soon as a mist ache is maid
 It nose bee fore two long
 And eye can put the error rite
 Its rare lea ever wrong

Eye have run this poem threw it
 I am shore your pleased two no
 Its letter perfect awl the weigh
 My chequer told me sew

---Anon

While automation can help improve your writing, ultimately there is no substitute for people (you know: actual *humans*) reading your CV, and the more people that read it the better. This could be potential employers, your critical friends shown in figure 6.20 or just reading it **out loud** to yourself described in section 4.5.1.

6.7.8 Your references

You might be tempted to put your referees details on your résumé. Don't bother because;

- references waste valuable space. You can say much more interesting things about yourself than who you referees are
- references aren't needed in the early stages of a job application anyway. Employers typically ask for your referees *much* later, when you've been offered or are about to be offered the job
- references disclose personal information. Do you really want to be giving personal details out to anyone that reads your CV? It could easily be misused.

It's not even worth saying **references available on request** - that just wastes space as well and is implied information on every CV anyway.

6.8 Breakpoints

Let's pause here. Insert a breakpoint in your `code` and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

- How long is your CV? How long should it be?
- One column or two column layout?
- How long should your personal statement be your CV, like Mike Rokernel has for example?
- Should you put education or experience first? Which is most important?
- How many of my hobbies and personal interests should you list? (Cheary, 2021)
- How can you beat the black hole mentioned in section 6.2? See Your Résumé vs. Oblivion (Weber, 2012)
- How many employers actually read cover letters?

*** RESUME**

6.9 Checklist: Big Bad Bugs

Here is a quick check-list for debugging your CV before you send it off to an employer:

1. Does it fit comfortably on exactly one page (*résumé*) or two pages (CV)? Definitely not one-and-a-half pages or more than two? See section 6.7.3
Does the style look good? Is it easy on the eye? Is there adequate whitespace, not too much (gappy) or too little (cramped)? See section 6.7.1
2. Have you used past tense consistently e.g. **developed** not **develop**, **collaborated** not **collaborate**
3. Is your year of graduation, degree program, University and expected (or achieved) overall degree classification clear? See section 6.6.2
4. Have you eaten your own dogfood (*woof*), see section 4.5.1? Is *everything* relevant? e.g. no swimming certificates from ten years ago?
5. Have you spell-checked using both automatic and manual (proof-reading) techniques? See section 6.7.7.
6. Have you got a second opinion from a “résumé robot”? Is it robot proof? See section 6.7.6
7. Have you shown you C.A.R.E? Are the **Contexts**, **Actions**, **Results** and **Evidence** described in section 6.6.7 clear? Have you added **context** using relevant hyperlinks that an interested reader can click on? See section 6.7.5
8. Is it in reverse chronological order with the most recent things **first**? Can your timeline be easily followed, with all dates clearly aligned for easy reading? See Neil Pointer as an example with a clear timeline using right-aligned dates.
9. Have you avoided using too many personal pronouns? **I**, **me**, **my** ... everywhere? See section 6.7.4
10. Have you made it clear what you have actually done using **prominent verbs**? See chapter 18
11. Have you given sufficient information on your education without going into too much detail? Have you mentioned courses you are studying now (*and next semester*)? See section 6.6.2
12. Have you **quantified** and evidenced the claims you have made where you can? See section 6.6
13. Is it balanced, including both technical and non-technical (softer) skills? See section 7.3.7
14. Does it have a good, clear structure? Not too many headings, around five sections for a one-pager see section 6.6.1?
15. Have you clearly distinguished between paid, unpaid and voluntary **experience**? Have you done the same for your **projects**, see section 6.6.4? Have you included *all* of the relevant experience that you can fit

- on including casual work, see section 7.3.7?
16. Has your CV been reviewed by other people? Do a CV swap with a critical friend (see figure 6.20) and score each others CV's using this rubric. This is a bit like pair programming. According to Linus's law "given enough eyeballs all bugs are shallow" (Raymond, 1999) so the more people who give you feedback the better
 17. Have you reviewed other people's CV's? This will put you in the shoes of an employer or recruiter, thereby helping you to write a better CV yourself. See the examples in chapter 15. Who would *you* want to interview and why?

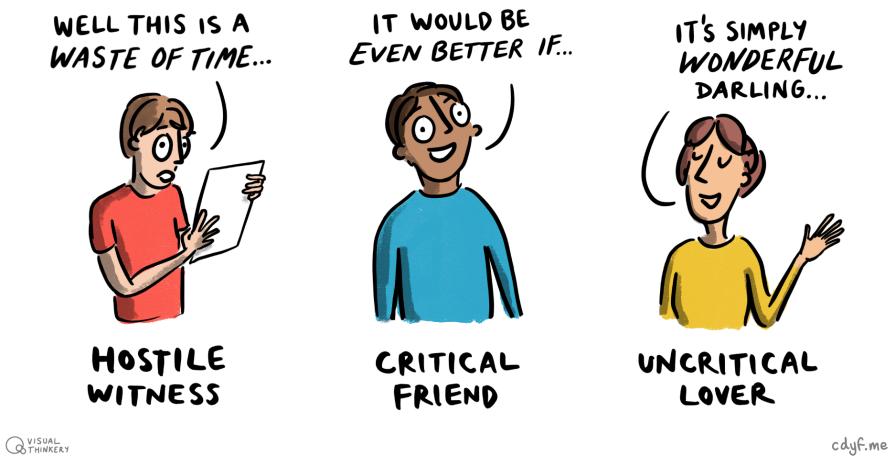


Figure 6.20: Showing your CV to somebody else is one of the best ways to debug it. You need to find a critical friend, somebody who won't just tear it apart (**hostile witness**) or tell you it's simply *wonderful* darling (**uncritical lover**) but tell you how to improve it, whatever state it is in (**critical friend**). Critical friend by Visual Thinkery is licensed under CC-BY-ND

6.10 Covering letters & personal statements

Applications and CV's are often accompanied by covering letters or include some kind of personal statement. They are your elevator pitch shown in figure 6.21. Whereas a lot of your CV is essentially a bulleted list of facts, numbers and evidence, a covering letter or personal statement gives you a chance to *really* demonstrate your fluent written communication skills in clear prose. If you're going to include a **personal statement** or **profile** on your CV keep it short, unlike Mike Rokernel who waffles on for ages without providing any evidence. This kind of information usually fits better in your covering letter anyway, because there isn't much space for it on a CV.

An elevator pitch quickly summarises an idea, product or service during a short journey in an elevator

Elevator pitch



Figure 6.21: Covering letters and personal statements are your elevator pitch. They should summarise why you're applying and what you have to offer, in a way that persuades the reader (or listener) that they should invite you to interview (before they get out of your metaphorical elevator). Elevator image by Downtowngal is licensed CC BY via Wikimedia Commons w.wiki/5wGF adapted using the Wikipedia app

Let's say you're applying for a widget engineering position at the world famous gadget company `widget.com`. There are three things you need to convey in the following order:

1. **Why them?** Why are you applying to `widget.com`
2. **Why that role?** Employees of `widget.com` have many roles and responsibilities but what is it about `widget engineering` (say) that attracts you?
3. **Why you?** Why should `widget.com` employ you? What skills and experience make you stand out from all the other candidate widget engineers? What are your Unique Selling Points (USPs)? Say why *you* would be good for *them* (**not** why `widget.com` would be good for you and your career).

You can see some examples of answers to these questions at:

- prospects.ac.uk/careers-advice/cvs-and-cover-letters/cover-letters/sample-cover-letter
- targetjobs.co.uk/cv-and-cover-letter-templates
- www.careers.manchester.ac.uk/applicationsinterviews/cl

6.10.1 Does anyone actually READ covering letters?

Some employers will read your covering letter very carefully, others less so. It is not always clear which employers will bother and which won't.

Even if *nobody* reads your covering letter, it is still worth writing one because formulating answers to the three basic questions in section 6.10 will force you to rehearse standard interview questions.

Think of your covering letter as practicing some of the lines of your elevator pitch shown in figure 6.21.

6.11 Debugging summary

Too long, didn't read (TL;DR)? Here's a summary:

In this chapter we have looked at how to debug your future, with a particular focus on your CV (or résumé). If you fix the bugs we've described here, before an employer sees your CV, you'll stand a much better chance of getting an interview. The checklist above in section 6.9 is a good place to start.

Some of the strategies you apply to debugging your *actual* code, can also be applied to debugging your future:

- Inspect, don't squash bugs, take time to reflect (see section 11.6) and *understand* any bugs you find in your future
- Being stuck is temporary, see section 14.10
- Check your assumptions, especially if debugging the `content` bugs described in section 6.1 makes you anxious or depressed, see chapter 3

- Don't go it alone, grow your network, see section 8.2.5
- Build your debugging toolkit by identifying your strengths and weaknesses, see chapter 2

Debugging your future can be an enjoyable adventure! Debugging your future is Coding your future.

In the next part (chapter 8) we'll look at using your CV and other written applications to help you *find* your future.

Chapter 7

Experiencing your future

So, tell me, are you experienced? Why is experience valuable and what kind of experience are employers looking for anyway? How can you get some more experience?

7.1 What you will learn

By the end of this chapter you will be able to

1. Describe why having experience can improve your chances of getting interviews
2. Identify what counts as experience and why it's valuable
3. Recognise opportunities to get more experience before you graduate

7.2 Why is experience so valuable?

You are *much more* than your grades, because your education is only a part of who you are. You are the sum total of your experiences, this is one of the reasons that experience is so valuable, see figure 7.3. It's common for students to be focused on their grades, whether those grades are low, middling or high. At the extremes, if you have got lower grades than you'd like, you might be anxious or unhappy about them. If you've got higher grades, you're probably focussed on keeping them high. While its important to focus on getting the best grades you can, employers are interested in the whole you rather than narrowly focusing on your academic education. Alongside your academic achievements, there is also your:

- Experience paid work, see section 6.6.3
- Projects of all kinds, see section 6.6.4
- Leadership and awards if you have or can demonstrate any 6.6.5



cdyf.me

cdyf.me

Figure 7.1: Do you respond with a sheepish `experience not found` when people ask about your experience? Is your experience like the classic page not found error message HTTP 404? The client sent you a valid request for your experience, but your server couldn't find it. Awkward. Embarrassing silence? Don't worry, there are some simple and easy ways to build your experience so that instead of negative 404's, you can respond with a cheerfully positive 200 (OK), as described in this list of HTTP status codes. We'll look at some of them in this chapter. Experience not found sketch by Visual Thinkery is licensed under CC-BY-ND

cdyf.me

cdyf.me

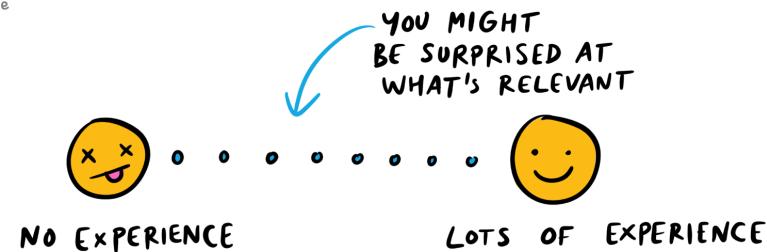
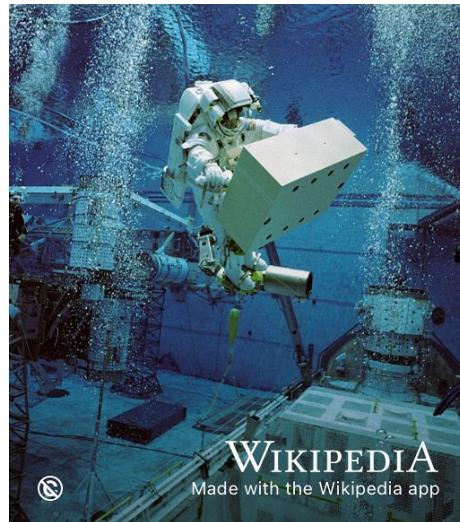


Figure 7.2: You might be surprised by what kinds of experience are relevant on your CV. There are lots of experiences beyond paid work such as projects and extra-curricular activities you've been involved in that tell a story about who you are and what you are capable of. What's relevant sketch by Visual Thinkery is licensed under CC-BY-ND

"experience" generally refers to know-how rather than descriptive knowledge (or in other words, on-the-job training rather than know-what or facts)

Experience



WIKIPEDIA
Made with the Wikipedia app

Figure 7.3: Experience is one of the best ways to develop *know-how*. While your formal education and academic study can help you develop *know-what*, you need to complement this knowledge with a range of experiences and on-the-job learning. This astronaut is training to work in microgravity by completing tasks underwater in a space suit. Like the astronaut, your education needs to combine academic study, with practical experience on-the-job. Public domain image of Christer Fuglesang training in the Neutral Buoyancy Laboratory (NBL) by NASA on Wikimedia Commons w.wiki/3WBf adapted using the Wikipedia App.

Your experience tells a story about who you are, what you're capable of and what you have learned. Experience is a crucial part of *how* you learn and allows you to demonstrate *what* you have learned. Experience makes you more *employable* which means:

1. experience tests and validates what you have already learned, see Sneha's story in chapter 25
2. experience motivates you to learn more by helping you decide what to learn next
3. experience improves your confidence (Carter, 2021)
4. experience broadens and deepens your skills and knowledge, see figure 2.1
5. experience improves your chances of being invited to interview
6. experience improves your chances of being offered a job after an interview
7. experience builds your professional network of contacts, particularly the crucial weaker ties described in section 8.2.5

Paul Redmond at the University of Liverpool describes experience as a key part of employability (Redmond, 2010) as expressed in his graduate jobs formula shown in equation (7.1).

$$E = Q + WE + S \times C \quad (7.1)$$

According to Redmond, your employability (E) is the sum of your qualifications (Q), your work experience (WE) and your strategies (S) multiplied by your contacts (C). It is difficult to quantify employability so precisely but Redmond's equation (7.1) is a good starting point for discussion. We will look at some strategies

- For qualifications (Q), see section 6.6.2 on your **Education**
- For strategies (S) and contacts (C) see chapter 8 on *Finding your Future*
- For experience (WE) read on, this chapter is all about your **experience**

7.3 Are you experienced?

So what counts as **experience**? I'm going to use experience to mean applying what you've learnt in your formal education outside of school or University, this includes but is not limited to, paid work. Employers use terms to describe jobs and experience for undergraduates and graduates inconsistently. So I've defined and outlined terms for relevant kinds of experience shown in table 7.1 and we'll use these definitions throughout the book.

7.3.1 Other experience

Some of the experience outlined in table 7.1 was probably what you were already thinking of as experience, however there are three important sources of experience that students often overlook:

Table 7.1: Are you experienced? Terms used throughout this guidebook to describe experience, employment and their definitions

Experience	Description
Casual	Part-time or casual work, for example in hospitality or retail etc
Voluntary	Unpaid, both in technical and non-technical roles for charity or nonprofit organisation
Social	Unpaid, participating in a club or society e.g. students union, sport etc
Entrepreneurial	Self-employment, freelancing, contracting, “moonlighting” in a [side job](https://en.wikipedia.org)
Insight	Usually no contract of employment. One to three weeks, sometimes known as work experience
Internship	Fixed term contract of employment, typically 3 months full-time over summer, but anywhere from a few days to a year
Placement	Fixed term contract of employment, typically 12 months long and an assessed part of a degree
Graduate job	Full-time permanent contract typically working in one department of an organisation
Graduate scheme	Full-time permanent contract. Fast-track or high-flier managerial scheme, in your first two years

1. **Voluntary work:** Any kind of work where you've given your time and labour to a community. That could be non-technical (working for a charity) or technical, such as contributing to open-source software, see section 7.3.4 and figure 7.4
2. **Casual work:** Working in hospitality or retail (etc) is often overlooked by students as an important source of relevant experience. It doesn't have to be technical to be relevant to employers, see section 7.3.7
3. **Tinkering:** having side projects as a playground to help you learn new skills and knowledge, either solo or collaboratively see section 7.3.6. For more competitive employers, side projects are likely to be more important. To get into ultra-competitive industries like gaming, candidates are often expected to have a portfolio of one or two games developed outside of University
4. **Student societies** Your students' union will have hundreds of official societies you can get involved in, and they'll be plenty of unofficial fringe communities too. As well as helping you develop new or existing interests, these societies give you an opportunity to serve a particular community of interest. Many societies seek members to take on positions of responsibility, above and beyond merely participating in their events. They provide fantastic opportunities to build new skills in a safe and supportive environment.

Before we discuss these experiences, lets look at some of the more conventional places for getting experience:

7.3.2 Big name experience

It's probably easier than you might think to get a big tech or big employer name on your CV. For example, many large employers run insight days, vacation schemes and spring weeks. These are often aimed at first year undergraduates,

an individual or group freely giving time and labour for community service

Volunteering

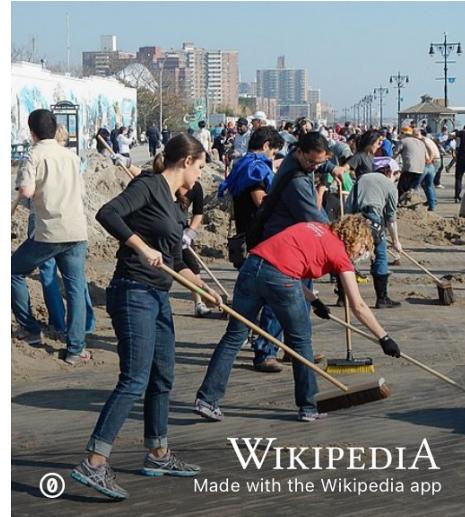


Figure 7.4: Volunteering is a great source of experience that employers value. That could mean volunteering for charitable causes, taking responsibility in a student society or getting involved in open source projects. Picture of volunteers cleaning up after Hurricane Sandy in 2012 by Jim Henderson via Wikimedia Commons w.wiki/3Z96 adapted using the Wikipedia App.

and are sometimes less competitive to get into than a longer term commitment such as a summer internship, year-long placement or even graduate job. A big name on your CV early in your degree can help it stand out later, as fluff bucket the grinning cheshire cat demonstrates on their CV shown in figure 7.5.

Other ways to get a big name on your CV include joining a big name competition or event, for example:

- Amazon hosts the Alexa challenge, see developer.amazon.com/alexaprize
- Apple hosts the Swift Student Challenge developer.apple.com/wwdc22/swift-student-challenge/
- The European Space Agency (ESA) organises the CANSAT competition (satellite in a can) see esa.int/Education/CanSat and cansatcompetition.com (NASA)
- Facebook has hackathons, see facebook.com/hackathon and developers.facebook.com
- Google hosts several events including:
 - Code Jam, HashCode and Kick Start codingcompetitions.withgoogle.com
 - Summer of Code summerofcode.withgoogle.com, see figure 7.6 (Googler, 2022)
 - Developer Student Club Leads developers.google.com/community/dsc/leads
 - Inside Look buildyourfuture.withgoogle.com/programs/inside-look



Figure 7.5: It's easier than you might think to get a big name on your CV, sometimes these can help your application stand out from the competition. Big name sketch by Visual Thinkery is licensed under CC-BY-ND

- Outreachy.org provides internships for people subject to systemic bias and underrepresentation in the technical industry where they are living, and is backed by big employers.
- IBM hosts the annual Call for Code developer.ibm.com/callforcode unlike other competitions, these have a corporate social responsibility (CSR) themes for the benefit of society at large
- Microsoft hosts the Imagine Cup imaginecup.microsoft.com and also has hands on learning paths at learn.microsoft.com
- Besides these kinds of dedicated events organised by big names, there are many competitions which are sponsored by them, see section 7.3.6

Big names can look good on your CV, but they are not the only way to make your CV stand out.

7.3.3 Small name experience

Any experience will help your CV stand out. Smaller employers have the advantage that they tend to be less picky than big names so it is often easier to get a foot in the door. It might not be what you see yourself doing for long, but the experience gained in a small company can be invaluable. You might even prefer working in a small company to a big corporation or conglomeration.

**Students get paid to
write free and open
source software**

Google Summer of Code



Figure 7.6: Large well known organisations like Google have a range of schemes and competitions that allow you to get their name on your CV, without working for them directly as an employee. Google Summer of Code summerofcode.withgoogle.com is just one example, where students get paid to write free and open source software (FOSS). (Googler, 2022)

7.3.4 Open source experience

Open source software projects are a great way to get some solid experience of software engineering, see for example Why Computing Students Should Contribute to Open Source Software Projects. (Spinellis, 2021) There's two ways to get started:

1. Raise a new issue via the project's issue tracker, such as github issues (Octocat, 2020)
2. Fix a bug by picking existing issues. (Robertson, 2020) It might sound trivial, but fixing a bug demonstrates that you can collaborate with others, understand the architecture and toolchain being used (which might be complex) and solve problems. See firstcontributions.github.io and the <good first issue> tag which helps new contributors identify starting points, see goodfirstissue.dev for some aggregated examples.

There are lots of different motivations for getting involved in open source, shown in figure 7.7. Whatever your motivation, contributing to open source software is fun, you'll learn heaps and it will look *great* on your CV. Open source software is widely used by, so contributing is a great way to get some real world experience of software development. Many open source projects are funded by employers both large and small, and you can get paid to develop open source software through projects like Google's Summer of Code. (Googler, 2022)

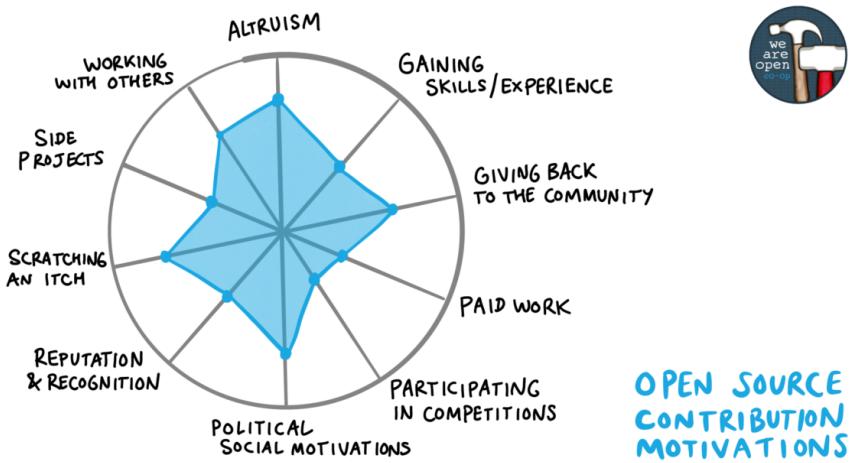


Figure 7.7: There are lots of good reasons for getting involved in open source software, gaining skills and experience of real software engineering in the wild is just one of them. Open Source Motivations by Visual Thinkery is licensed under CC-BY-ND

If you're looking for a project to get stuck into, here are Diomidis Spinellis tips for getting started (Spinellis, 2021):

1. Choose a project with several active contributors, so that there is a community to help you
2. Choose a relatively popular project (with some GitHub stars) so that you can avoid abandonware but...
3. ... Avoid “blockbuster” projects like tensorflow or vscode , so that your contributions will not get lost in the politics and bureaucracy of a large project
4. Verify that you can build and run the project from your own setup
5. Ensure the project regularly accepts pull requests from outsiders, so that your contributions will have a chance of being accepted
6. Contribute a trivial fix to start with to test your ability use the project’s workflows

The guidelines for prospective contributors to Google’s Summer of Code (GSoC) at google.github.io/gsocguides/student go into more depth about how you can get involved in open source software development. Many of these general guidelines apply both inside (paid) and outside (unpaid) of GSoC. (Googler, 2022)

7.3.5 Voluntary experience

Experience as a section of your CV usually means *paid* work. However, experience in the context of this chapter means anything where you can show you've been part of a bigger team, taken responsibility for something or tried to make the world a better place somehow. These include:

- Volunteering: Doing voluntary work is a good way to pick up new skills
- Being involved in societies: e.g. taking responsibility for things in a society
- Getting involved in a community, either physical or online

7.3.6 Hacking & tinkering experience

Another good source of experience is tinkering or hacking. Hackers and tinkerers improve or repair things in their spare time for fun, either as a solo project or collaboratively. For example, why not build an experimental:

- Home media centre?
- DIY smart energy meter?
- Plant watering system for when you go on holiday?
- See also github.com/coderafters-io/build-your-own-x for inspiration

Yes, these projects are amateurish, but tinkering and hacking will clearly demonstrate your passion for learning and engineering. Apart from being fun, weekend hacks can look *great* in the **projects** section of your CV, see section 6.6.4.

an event where people engage in rapid and collaborative engineering over a relatively short period of time such as 24 or 48 hours

Hackathon



Figure 7.8: Hackathons are collaborative social events where you can learn how to rapidly prototype ideas in a friendly and supportive environment. Picture of participants at a Wikimedia Hackathon in Prague by Chris Koerner on Wikimedia Commons [w.wiki/5rjR](https://commons.wikimedia.org/w/index.php?title=File:Wikimedia_Hackathon_in_Prague_2014_group_picture.jpg&oldid=10731111) adapted using the Wikipedia app.

Attending Hackathons is another way to learn and gain **experience** by tinkering, in a collaborative and social environment, see figure 7.8. The word hackathon is a portmanteau of hacking (from hacker culture not security hacking) and marathon. Hackathons tend to be short (24 to 48 hours) but intense, see for example:

- hackathons.org.uk
 - greatunihack.com
 - studenthack.com
- devpost.com/hackathons
- mlh.io (Major League Hacking)
- hackerearth.com
- hackzurich.com

Alongside these competitions, there are other hackathon-type events which are organised by a single employer, see section 7.3.2. There are also opportunities to build your technical experience by practicing coding interviews online such as hackerrank and leetcode, described in section 10.2.3.

Side projects like these help to distinguish you from your peers, but they don't *have* to be technical, see section 7.3.7 and 7.3.5. You might feel that you are too busy to do any side projects, but if working hard on academic study and getting top grades means that you:

- Neglect hobbies, interests and friends
- Start to suffer from anxiety or depression
- Only have **projects** that were a compulsory part of your formal education

It might be time to reflect and consider re-balancing your priorities if you can.

7.3.7 Casual experience

You may already have experience of paid employment as a casual or part-time worker. This could include jobs such as waiting tables, serving in a bar or working in other areas of hospitality or retail, for example as a Saturday job.

It is important to recognise that these jobs have value. Many students make the mistake of overlooking their casual work experience because they discount it as non-technical and "low-skilled". In section 6.6 we saw that one of the stories you want to tell in your job applications is that you:

1. take responsibility
2. achieve things
3. are nice to have around

Doing casual work can demonstrate all of these things. For example, from the ages of 12 to 18 I was a paperboy, except unlike the Titanic paperboy selling newspapers in the street in figure 7.9, I delivered newspapers directly to the doors of paying customers every morning. This was not a particularly highly skilled job, but it *does* demonstrate:

A paperboy is someone – often an older child or adolescent – who distributes printed newspapers

Paperboy

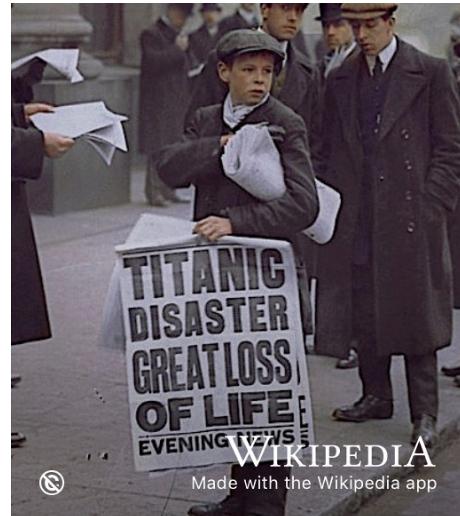


Figure 7.9: Casual and part-time work tell an important story about you on your CV. For example, from the age of 12, I was a paperboy, delivering newspapers directly to the doors of paying customers. This demonstrates reliability and work ethic, because I did this in all weathers (sun, wind, rain, snow, hangovers etc) for six years. If you have casual experience like this, don't forget to include it in your CV. Public domain image of the Titanic paperboy, Ned Parfett selling newspapers in London via Wikimedia Commons at w.wiki/35HA adapted using the Wikipedia app.

1. work ethic: getting up early *every* morning (including Saturdays). Sometimes work is about just turning up everyday!
2. taking responsibility and being reliable
3. understanding the value of money by earning a wage

I've often spoken to students who neglect to tell me about their paid work in retail or hospitality. "But it's not technical" they say, "it's low skilled and irrelevant". However, serving customers demonstrates your ability to provide good customer service and work as part of a team, often under pressure, see figure 7.10. This is good evidence of the "nice to have around" bit that Jonathan Black refers to (Black, 2019b) and is something your formal education will not typically provide much evidence of. So, don't fall into the trap of discounting the value of casual or part-time labour.



Figure 7.10: Early in your career, casual work in hospitality or retail, such as a supermarket like Budgens where I used to work as a teenager, is worth mentioning on your CV. If you have any experience of this kind, make sure you mention it and describe the skills you developed. Think carefully about the verbs you can use to describe casual experience, see chapter 18.

7.4 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

- What experience do you have to date?
- What activities could you do to get some more experience?

- What are the pros and cons of summer internships vs. year long placements?
 - Which one is right for you?

* RESUME

7.5 Summarising your experience

Too long, didn't read (TL;DR)? Here's a summary:

Experiencing your future is coding your future.

This chapter is under construction because I'm using agile book development methods, see figure 7.11.

The Death Star is a
fictional mobile space
station and galactic
superweapon featured in
the Star Wars space-
opera franchise

Death Star

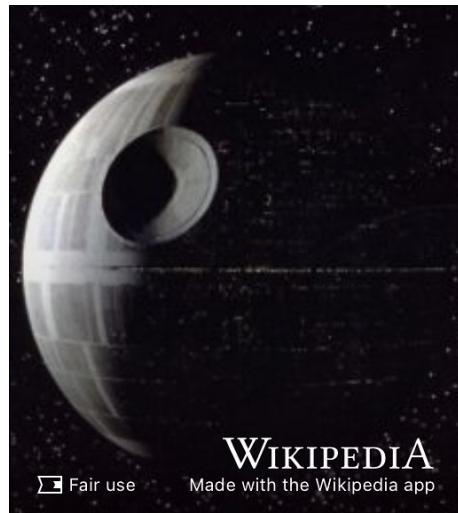


Figure 7.11: Just like the Death Star, this galactic superweapon book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Chapter 8

Finding your future

So, you've successfully debugged your CV, with help from chapter 6. You've reflected on and identified where you can improve your **experience** as described in chapter 7. Now, how can find an interesting job? How can use your CV, covering letter and any other communication to persuade employers to invite you to an interview? What techniques exist and how can you use your networks to help you? Where can you look? This chapter will help you find your future.

8.1 What you will learn

At the end of this chapter you will be able to:

1. Formulate job search strategies:
 - by role, sector, skills, time, size, quantity, values, culture, network, salary and location
2. Apply your search strategies to advertised (and unadvertised) opportunities
 - Identify opportunities for finding work, online and face-to-face
 - Identify people in your existing networks who can help you
 - Grow your networks and use them to your advantage
3. Describe some of the problems with recruitment:
 - for employers
 - for potential employees (like you)
 - Critically evaluate what employers have on offer (beyond the financial incentives)



Figure 8.1: Coding your future is all very well, but how do you actually get a job? This chapter looks at job searching and networking. Yes but... sketch by Visual Thinkery is licensed under CC-BY-ND

8.2 Job search strategies

Before we look at where to look for jobs, we need to discuss what to look for. Let's imagine there is a handy AI-powered function in python called `find_jobs()` which knows *everything* about:

- all the jobs available, regardless of where they are advertised, including hidden ones
- all of your skills, knowledge, values and ambitions
- every employer in the world, their services, products and values
- everyone in your social and professional networks, right from first degree connections through to sixth degree connections (Bacon, 1994)

Such a function doesn't really exist yet, but let's pretend that it does to illustrate some points about job searching. If you were to run the function with no arguments...

see <https://gist.github.com/dullhunk/6f3a98ca6ae2cdcb611f8b4838fe25df>

...you will be completely overwhelmed with the results. It can be bewildering knowing where to start your job search, because the possibilities are endless. Your search space is huge.

Thankfully, we can break it down, but there are many parameters and variables in your search that you need to think about. Here are some basic strategies to help you start to decompose the complex problem of job hunting into smaller, more manageable problems you can start to tackle. We've already mentioned some:

1. by role, see section 8.2.1
2. by sector, see section 8.2.2
3. by skills, see section 8.2.3
4. by location, see section 8.2.4
5. by network, see section 8.2.5
6. by salary, see section 8.2.6
7. by size, see section 8.2.7
8. by values, see section 8.2.8
9. by timing, see section 8.2.9
10. by quantity, see section 8.2.10

Each strategy has its own strengths and weaknesses, so you'll probably want to combine more than one and experiment with different strategies as your search progresses over time. No doubt, you'll have some extra strategies of your own, this is not meant to be an exhaustive list, just a starting point:

8.2.1 By role

There are many different roles that studying computing gives you access to besides software engineering. So one strategy is to research the roles described

in section 9.2, focus on some you like the sound of and then apply for that kind of role. Let's say you are interested in **software engineering**. This is a strategy that will help you narrow down a huge number of jobs into something more manageable. For example:

see <https://gist.github.com/dullhunk/c13ddb4dbd5c43220cd35caa5d1f40c7>

The list of roles in section 9.2 will give you a flavour of some roles to think about, beyond software engineering.

8.2.2 By sector

Computing is found in every sector of business and society. This means won't just find computing jobs in the technology sector, whatever that is. Although the lines between some sectors are becoming increasingly blurred, you can still narrow down the options by picking a sector you are interested in and pursuing that. For example:

see <https://gist.github.com/dullhunk/ecc64065d98a45b54ed178f370af6abf>

Like many graduate employment guides, the Guardian 300 guide (see section 8.3.1) has good overview of different sectors by breaking down employers by sectors such as:

- Consulting
- Public sector
- Finance

Picking a sector can help you get started, by narrowing what might otherwise be a massive search space. The sector(s) you choose will also determine the kind of salary you can expect, see section 8.2.6. (Tennant and Lowe, 1986)

8.2.3 By skills

Is there a specific skill you have that you enjoy using, or a technology you know that has a bright future? You could use this as a way to find jobs, for example:

see <https://gist.github.com/dullhunk/61bf8c20b4eb2da9a168ebb29b7026fd>

Picking a specific technology can sometimes help you narrow down the options, for example by using these keywords in your search.

8.2.4 By location

Pick a location. London? Manchester? Paris? Geneva? New York? Silly Valley? Depending on the location, this will determine the options that are available:

see <https://gist.github.com/dullhunk/5004cbf5fc337fb8e7357e62c50d15eb> ... and so on. So locations can help you define (and potentially reduce) your

problem space. For example, if you're looking in Manchester, you can get started at git.io/manc. Picking a location can also help you identify networking opportunities for hidden, un-advertised vacancies and making speculative applications. You could find out about employers by attending free [meetup.com](https://www.meetup.com) or [eventbrite.com](https://www.eventbrite.com) events in your chosen location.

8.2.5 By network

The people in your network can help you find work. *It's not what you know, it's who you know*, or so the cliché goes. (Spearpoint et al., 1992) The bigger your network, the more opportunities you'll know about. So a really basic strategy for finding jobs is using your networks. Remember, `find_jobs` does just know what you know, it also knows who is in your professional `network` and the employers *they* know about, so we could ask it to find opportunities in your professional and social networks:

see <https://gist.github.com/dullhunk/c2e05b55e55c285de8525a6952731f30>

Your close network probably won't change that much, the friends and family you trust and rely on. Its important to recognise the importance of more casual acquaintances, or what sociologist Mark Granovetter calls "weak ties". (Granovetter, 1973)

Weak ties are people you don't know as well, but are important for a range of reasons. Research has shown that building networks of weak ties is good for your mental health and can give you an edge in job hunting. (Leslie, 2020) Granovetter showed that many job opportunities came through weak ties, rather than strong ones. This is true not just of jobs early on in your career (like now) but also later too. So it is in your interests to continually foster weak connections and be open to serendipitous meetings where you bump into people, as in Figure 8.2. "Bumping into" here, could mean either physical or virtual.

8.2.6 By salary

You can also define your search space by salary:

see <https://gist.github.com/dullhunk/28c308ee12c067b95dd724eabfc9e291>

Setting a minimum salary will include or exclude certain employers from your search. To do this realistically, you need to know how much you are worth, see section 8.4.5 and figure 8.4 for some places to start.

8.2.7 By size

Do you see yourself working for a large multi-national corporation or a bedroom startup? Or something in-between? There's advantages and disadvantages to each, but like all our other parameters, picking an organisation size can help you get started. In some cases, size will also define the `type` of organisation, for example:

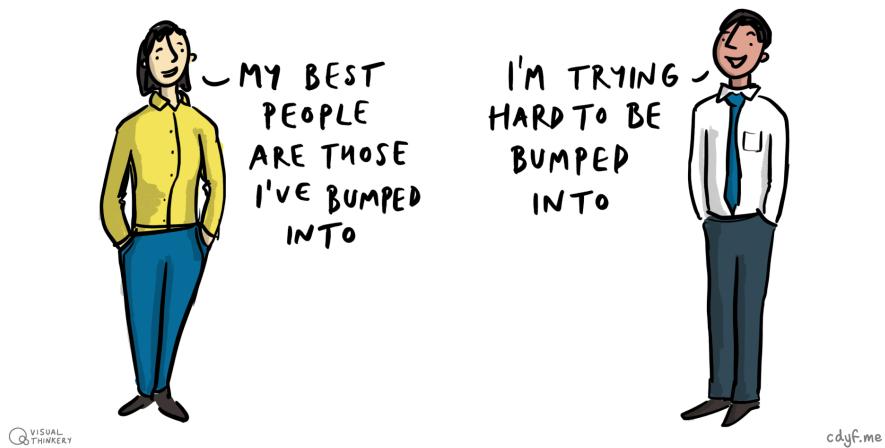


Figure 8.2: It's not what you know, its who you know: Networking and personal contacts can be more useful than just knowledge and skills alone, when seeking employment. Networking is an essential part of any job search, your networks can help you now and in the future. One of the things looked at in this chapter is how to build and use your networks to help find the job you're after. The simplest networking technique is bumping into people, but you need to create opportunities for that to happen. Bumped into sketch by Visual Thinkery is licensed under CC-BY-ND

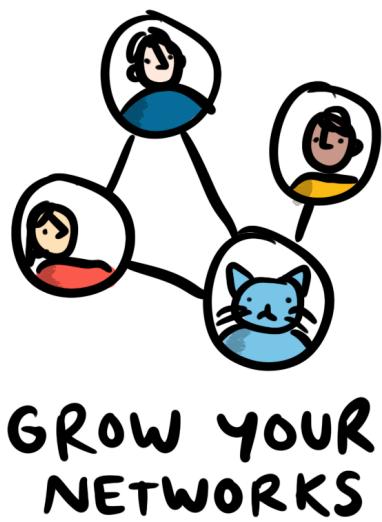


Figure 8.3: Who is in your network? Grow and use your network, both the strong ties and the weak ties. Weak ties are often the most important when it comes to job hunting. Networks sketch by Visual Thinkery is licensed under CC-BY-ND

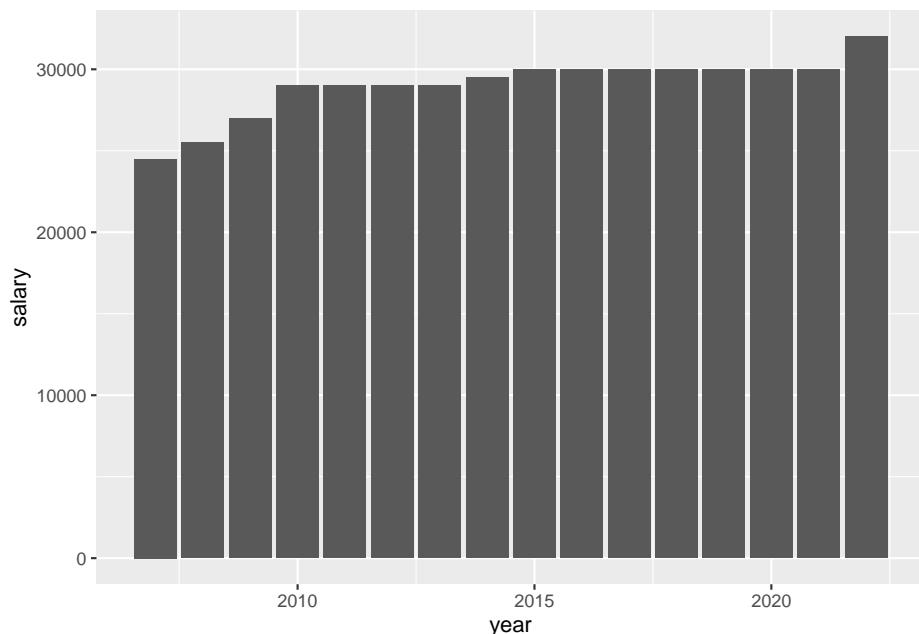


Figure 8.4: Average graduate starting salaries (in £) at the UK’s “top” employers 2007 to 2022, according to [highfliers.co.uk](#) (Birchall, 2022). Since 2007, graduate salaries have risen from £24,500 to £32,000k in 2022. Note that these are *graduate* salaries, not internship or placement salaries which tend to be lower. Also note that “top” employers is a highly subjective judgement, lots of good smaller employers don’t feature in this data and they typically pay less than the large multinational organisations dominating this sample. The averages also hide significant variations by sector and location, see the salary resources in section 8.4.5

Table 8.1: The practicalities of applying for vacancies, internships, placements, graduate jobs and schemes in large multinational employers and small to medium sized enterprises (SMEs)

	Large employers	SMEs
Where	Advertise broadly	Less likely to advertise on big jobs boards
When	Vacancies open earlier in the academic year	Vacancies *tend* to open later in the academic year
How	Typically multistage applications, several rounds of interviews	Typically shorter application and interview process
Who	Typically receive high volume of applications per vacancy	Typically receive lower volume of applications per vacancy
Process	Unlikely to consider speculative applications	May consider speculative or informal applications

see <https://gist.github.com/dullhunk/5ec7d37a60637c7083dc661e74ddcaf9>

Likewise, if you know you want to work for a multi-national, that gives you a list of employers you can target. Picking a size, can help you identify employers you would like to work for. Table 8.1 summarises some of the when and where some employers typically advertise.

8.2.8 By values

Besides the necessary value of making a profit or balancing the books, which is all employers need to do to survive, what are the values of the employers you are interested in? How do they match your own, the `values_match` below? Reasonably or not at all? How much of your soul do you need to sell? All of it, or just some of it? This a difficult question to compute, see section 9.5, but let's imagine we can:

see <https://gist.github.com/dullhunk/015a1182225f13c9307538468620ceb9>

An employers `values` are often reflected in the organisations `culture`. What is the employers culture like? What sort of structure would you fit into, see figure 8.5? Would you enjoy working there and would you feel like you belonged? How can you actually find this out? Do you need to sell your soul? If you're struggling with these questions, you may need to do some more soul searching, see section 9.5 and chapter 2. Talk to employees too, this is why networking is important, see section 8.2.5.

Our `find_jobs()` function is useful, if only as a thought experiment, but there's still the strategic questions of *when* you use the function and *how often* so lets look at those next.

8.2.9 By timing

For graduates and undergraduates, the time of year you are looking can determine what jobs you can look for. At the beginning of the academic year in September you might target large multinational organisations. If you're not

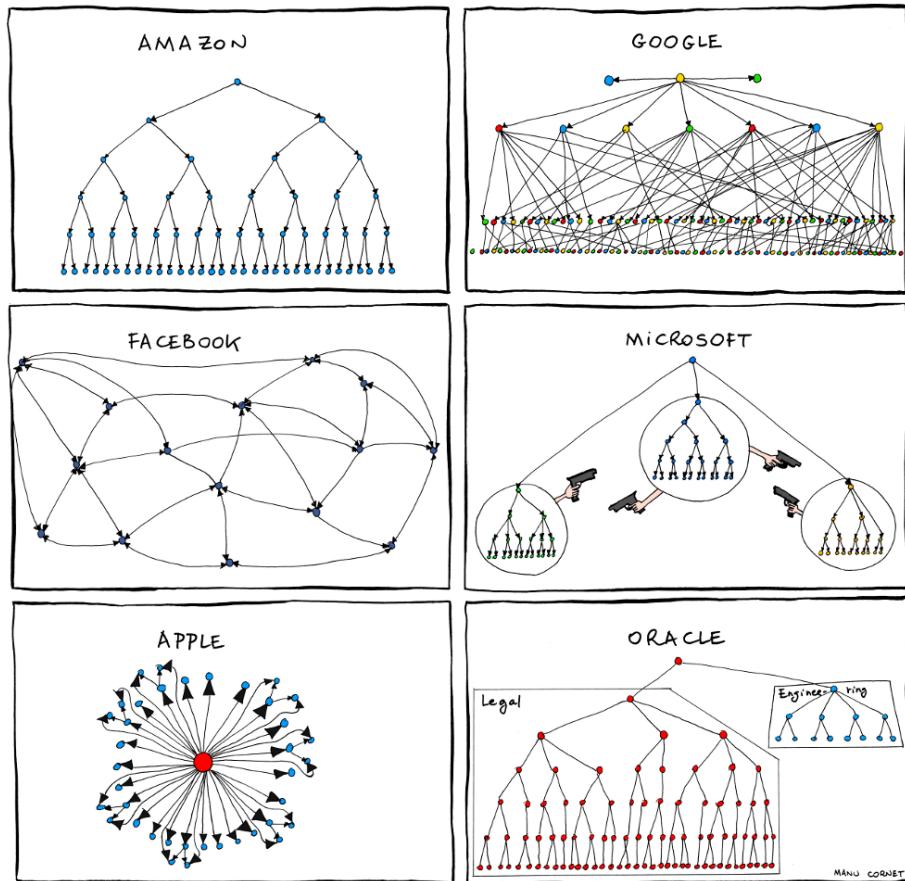


Figure 8.5: Culture and values are often reflected in an employers structure. Finding out about an employers structure will help you begin to understand their **culture** and **values**, see section 9.5. Creative Commons Attribution-ShareAlike org charts comic by Manu Cornet on Wikimedia Commons w.wiki/5jV8

successful, you could switch to smaller employers later in the academic year. For example “It’s October, so I’m applying to large employers” or:

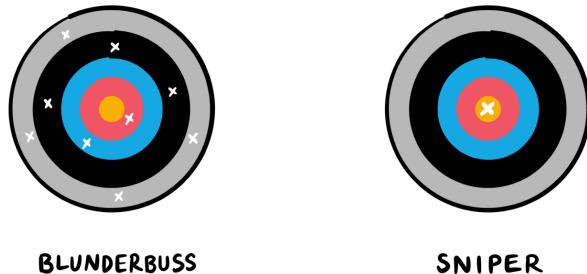
see <https://gist.github.com/dullhunk/83030d66f53da3d664f6cd3ed7746a36>

Armed with knowledge of *when* to apply you can start to target employers. Timing is important because in some cases it will sometimes determine the type of organisation you apply to.

8.2.10 By quantity

Should you optimise the *quantity* of your applications or the *quality*, see figure 8.6?

CHOOSE YOUR STRATEGY...



cdyf.me

Figure 8.6: As you target employers, what will your strategy be? At one extreme you could optimise the *quantity* of your applications, aiming to do as many as you can. This is shown in the left of the picture by the blunderbuss (or scattergun) strategy. You make lots of applications but don’t target or tailor them much in the hope that *some* will hit the target if you point your weapon (CV) in approximately the right general direction. At the other extreme you could optimise the *quality* of your applications by spending more time researching the employer and carefully aiming your shots more like a sniper would (in the right of the figure). Which strategy is best? Blunderbuss sketch by Visual Thinkery is licensed under CC-BY-ND

Figure 8.6 shows two extreme approaches to job hunting, in reality you’ll probably want to strike a balance between *quality* and *quantity*. Finding jobs by quantity can set a pace and rhythm to your job search:

* "I will do one application per week during term time"

or set a time limit for what you're prepared to do:

* "I'll spend two hours per week finding and applying for jobs"

If you find yourself:

- Making a small number of applications, you may need to consider applying more widely and spending *less time* on each application
- Making a large number of applications, you may need to consider applying less and spending *more time* on each application

This raises the question, how big is a *small* or a *large* number of job applications, which isn't easy to answer. Doing fifty applications over a couple of months may mean the quality of each application is too low.

8.3 Where can you look for jobs?

Using our fictional `find_jobs` function we've described and discussed some basic strategies to get you started. Sadly no such function exists yet, so armed with a rough idea of what you're looking for, where can you actually look?

The marketplace for job searching and job hunting advice is incredibly crowded. Employers spend huge amounts of money on recruitment and this is reflected in the enormous range of job websites, which are often accompanied by advice on job hunting. There are four kinds of places you can look for jobs:

1. Undergraduate and graduate jobs boards, such as Gradcracker and your University 8.3.1
2. General jobs boards, such as Google jobs (and the rest) see section 8.3.2
3. Portfolio style, such as LinkedIn and Github which allow you to publish a public profile, see section 8.3.3
4. Your network, online and offline, see section 8.2.5
5. Recruiters, get a recruiter to help you, see section 8.3.4

8.3.1 For graduates and undergraduates

The following resources are specifically tailored to undergraduate students and graduates looking for jobs in Europe and beyond:

- gradcracker.com for engineering and technology students, you can filter e.g. by Computing/Technology jobs, from the publishers of the popular gradcracker toolkit
- ratemyplacement.co.uk is a leading UK job resource for undergraduates seeking placements and internships.
- targetjobs.co.uk graduate jobs, schemes and internships from the people behind The Guardian 300 top graduate employers

- milkround.com, placements and graduate positions from the people behind The Times Top 100 Graduate employers
- gradconnection.com, for university students and recent graduates
- graduateland.com, placements and graduate positions around Europe
- prospects.ac.uk, a jobs board accompanied by job searching advice
- InsideCareers.co.uk is good if you're looking for jobs in the financial sector
- for remote (WFH) opportunities
- Year in Industry if you're looking for a year in industry
- Your University careers service. University jobs boards are good places to look for opportunities that are specifically targeted at students of the University where you are studying.
 - if you're studying at the University of Manchester it's careerconnect.manchester.ac.uk (UoM login required)
 - waggle.cs.manchester.ac.uk for University of Manchester students looking for jobs in computing

8.3.2 For any job seekers

The following resources are aimed at *anyone* (not just undergraduates and graduates) looking for jobs *anywhere* in the world. Let's start with Google:

Google job search shown in figure 8.7 is a good starting point. It doesn't index *every* job listings site, see Google's job hunting service comes to UK (Kelion, 2018), but its a pretty good place to start.

- Google job search indexes jobs advertised by many of the resources mentioned in this chapter. You can use google job search use to find internships, placements and graduate jobs anywhere in the world, as well as saving vacancies and setting up job alert notifications by email. If you haven't used it already try the searches below. Unlike other sites, Google job search works by indexing embedded microdata structured using schema.org/JobPosting. Keywords like **job** and **intern** in an ordinary (vanilla) google search will trigger the job search product, as shown in the following examples:

-google.com/search?q=software+engineering+intern+manchester
-google.com/search?q=business+analyst+intern
-google.com/search?q=graduate+hardware+engineer
-google.com/search?q=graduate+software+job+london
-google.com/search?q=data+scientist+intern
-google.com/search?q=research+software+engineer+job

-Google job search is an impressive product, see grow.google/job-seekers but it doesn't index *everything*. If you're looking for a job AT google, they

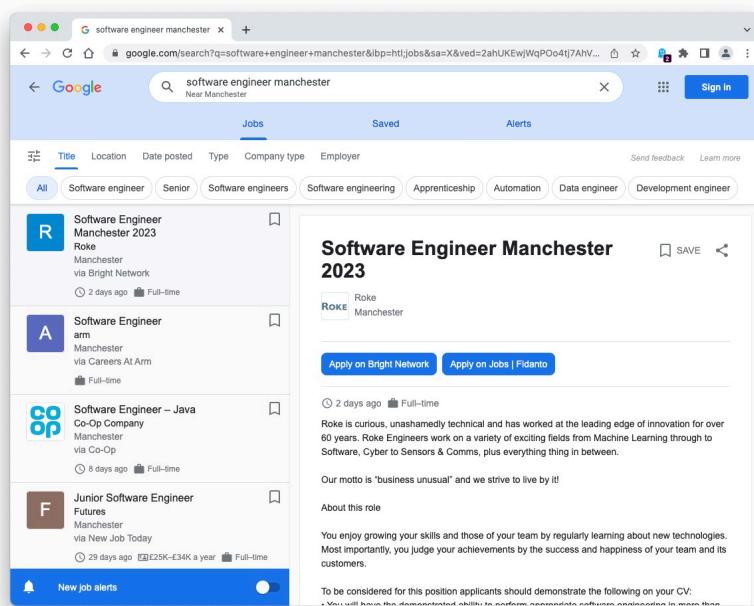


Figure 8.7: Keywords like `job`, `intern` and `engineer` in an ordinary (vanilla) google search will trigger Google's job search product, an enhanced search feature that aggregates listings from many different jobs boards. Here's an example from clicking through on the results of a search for `google.com/search?q=software+engineer+manchester`. You can refine results by `title`, `location`, `date posted`, `type`, `company type` and `employer` shown across the top of the screenshot, see the text below for more examples.

have moved from jobs.google.com to careers.google.com, see also section 7.3.2

- glassdoor.co.uk is like tripadvisor for jobs. Find out what it's *really* like to work for given employers from current and former employees. A student oriented version can be found at glassdoor.com/Students, this means you can use it without writing a review of a previous employer (which is what non-student users have to do to access the content)
- HiPEAC jobs (High Performance and Embedded Architecture and Compilation) is good for jobs in hardware, supercomputing and related fields
- Indeed.co.uk, adzuna.co.uk, cwjobs.co.uk, fish4.co.uk, reed.co.uk, totaljobs.com, monster.co.uk, jobs.smartrecruiters.com, workinstartups.com, cv-library.co.uk, jobs.ac.uk are general jobs boards that also advertise jobs for students and graduates, alongside many other vacancies.
- Otta.com for people with 0-10 years experience. From engineering to sales, discover jobs & internships at London's most innovative companies.

8.3.3 Pushing a public profile

Jobs boards use the **pull** approach to job hunting, you search for and pull vacancies of interest. Alongside this, you can also try a **push** approach by publishing a professional public profile online.

Publishing a portfolio of your work online will allow employers and recruiters to come and find you, as well as you searching for them. Employers shouldn't expect you to have an online portfolio and you might not want to have one, but it can help. If you choose to have a public profile, you should link to it in the header of your CV, see section 6.6.1.

- **LinkedIn.com** is a social media service which allows you to put professional details online a bit like a semi-public or public CV, see figure 8.8 and table 6.1. Besides allow you to advertise yourself, LinkedIn also:
 - advertises job vacancies linkedin.com/jobs
 - is frequently visited by recruiters
 - allows you to apply for jobs directly through LinkedIn
 - (although making fast applications is not always a good thing)
 - provides tutorials for students to get started linkedin.com/learning/learning-linkedin-for-students
- **Github.com** allows you to publish code in public repositories that everyone can see, see figure 8.9. If you're comfortable doing it, publishing your software online can be a good way of demonstrating your technical skills and knowledge but be careful publishing your coursework online, see section 6.6.4. Nothing says "*I can build software*" quite like "*here's one I made earlier*". (Blair, 1958)

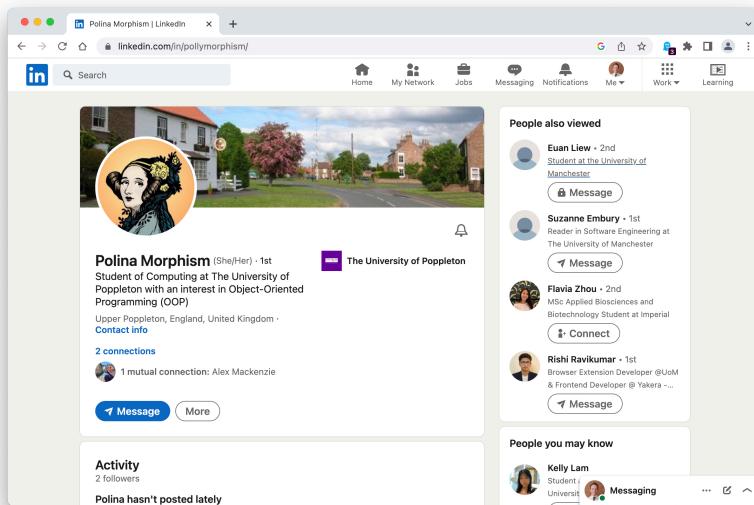


Figure 8.8: LinkedIn is a social media service which allows you to build a professional public profile and for employers to advertise their job vacancies. Social media caveats aside (see section 3.8), LinkedIn can be useful tool for networking with other professionals and finding a job, see table 6.1. You can see Polly's profile at linkedin.com/in/pollymorphism/

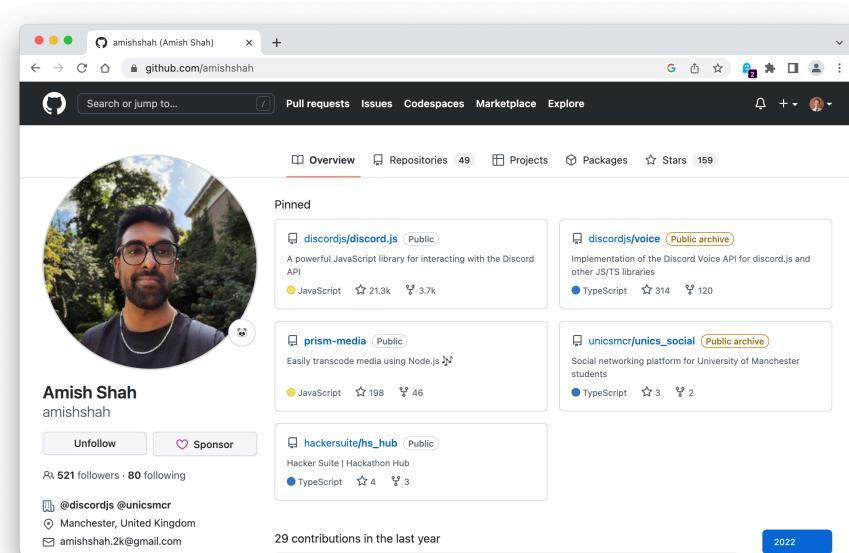


Figure 8.9: Github is commonly used to host open source software development projects. As of June 2022, GitHub reported having over 83 million developers and more than 200 million repositories, including at least 28 million public repositories. The screenshot shows a profile of Computer Science student Amish Shah: github.com/amishshah

- **Stackoverflow.com** You've probably already cut-and-pasted solutions from the question and answer forum Stack Overflow. You can also create a profile on Stack Overflow too, see figure 8.10. As of March 2021 the site had over 14 million registered users, and had received over 21 million questions and 31 million answers.

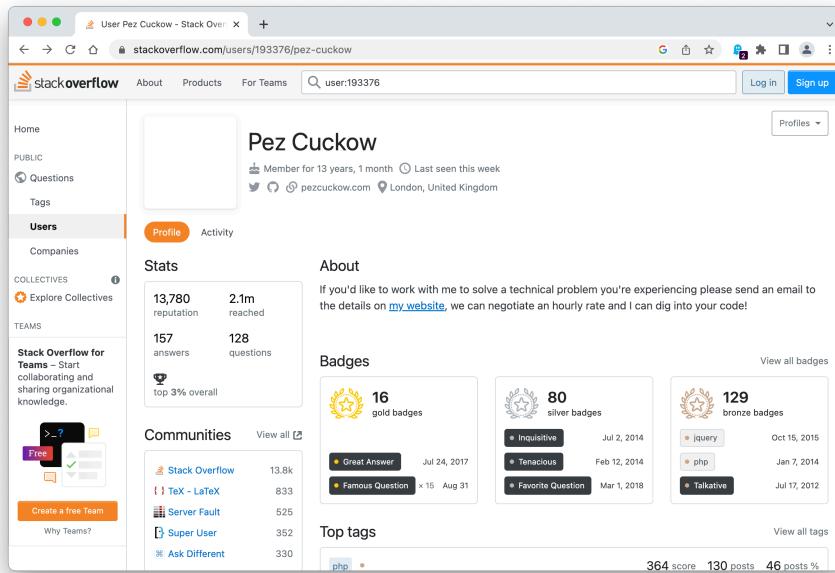


Figure 8.10: An example of using stackoverflow to create a public profile by asking and answering technical questions online. The screenshot shows a profile of former Computer Science student Pez Cuckow: [stack overflow . com / users / 193376 / pez-cuckow](https://stackoverflow.com/users/193376/pez-cuckow)

So, if you're happy to publish your work online, sites like LinkedIn, Github and Stackoverflow can help you find jobs using a *push* approach alongside more conventional *pull* approaches described in section 8.3.1 and 8.3.2. They can also be used to augment your CV, with the stuff that doesn't fit onto one or two printed pages. Digital profiles can be also be useful if you're looking for freelance, contract or part-time work.

8.3.4 What about recruiters?

Recruiters can help you find work and they operate in every industry sector. They are sometimes called “head-hunters”, and there are two basic kinds that can help you:

1. Recruiters employed directly by an employer, for example in the human

resources (HR) department of a given organisation.

2. Recruiters who are self-employed or work for a recruitment agency. They typically earn money from the number of interview candidates and successful hires they provide for their clients. They are often deliberately coy about who their clients are.

Recruiters are usually not technical people, so don't expect them to any knowledge of software engineering (for example) - that isn't usually their skill set. Although recruiters can help you, it is worth being wary of recruiters as shown in figure 8.11, especially if they work for an agency rather than being directly employed by the organisation you are interested in.



Figure 8.11: The autocomplete algorithm of a well known search engine gives you an idea of what *some* people think about *some* recruiters. This doesn't mean you should avoid recruiters completely, just be careful how you use them and pay attention to who they work for.

Some recruiters are very good and can help you. For example, there are some recruitment agencies that specialise in helping employers recruit graduates, these may be useful to you. However some recruiters are not very good, and don't provide a valued service for employers or potential employees like you. This is why you sometimes see no **recruiters** or no **agencies** on job adverts. So be wary of recruiters, and remember that some recruiters work primarily for their clients (employers) not you.

In most cases you shouldn't have to pay recruiters up front but job scammers will sometimes pose as recruiters so beware. Talking of job scammers, there's some things you need to be wary of when you are job hunting:

8.4 Buyer beware

When you're looking for job you're acting as both a buyer *and* a seller.

1. **SELLING:** You're selling your services in a marketplace, for the best price you can get
2. **BUYING:** You're buying into the culture and values of an employer (see section 9.5), who are trying to sell themselves to you.

As a buyer and seller, you should be wary of the following:

- Job scammers: section 8.4.1
- Over-specified jobs: section 8.4.2
- Unpaid internships: section 8.4.3
- Overselling: section 8.4.4
- Underselling: section 8.4.5
- Compromises: section 8.4.6
- Time sink: section 8.4.7
- Rejection: section 8.4.8
- The rollercoaster: section 8.4.9

8.4.1 Beware of the job scammers

Most job adverts are legitimate but you are vulnerable when you are job hunting. You may become more vulnerable over time if you are getting repeated rejections (remember: repeated rejection is quite normal). Unfortunately there are some shady characters out there looking to exploit your vulnerability through various kinds of employment fraud. (Hinds, 2017) You should be wary of anyone asking you for:

- Money up front - be very suspicious
- Excessive personal data such as your birth date, passport number and bank details. These could be used for identity theft, fraud or other criminal activities
- Suspicious contact details and generic non-work free email addresses, e.g. gmail, outlook.com etc (Smith and Rosser, 2021)
- See more examples in figure 8.12 and at google.com/search?q=job+scams

Reputable employers (and jobs boards) will not try to scam you, but you should beware of job scammers if you find yourself looking for employment off the beaten track. Like Pinocchio, it's quite easy to spot lies once you recognise some of their nosy signals.

8.4.2 Beware of over-specified jobs

Employers and recruiters routinely over-specify job descriptions. A good example of this is, when the Swift programming language was first publicly released in 2014 at the Apple Worldwide Developers Conference (WWDC) in California, job adverts instantly appeared asking for programmers with **5 years experience in Swift!** How can *anyone* have five years experience in a programming language that's only just been made public?! Aside from the people who developed the language, like Chris Lattner, the recruiters and employers must have had a *long* search trying to find their candidate. It must have taken them at least five years!

The moral of this Apple story above is, if you don't meet *all* the criteria in a job specification, that shouldn't stop you applying. Most job specifications



Figure 8.12: Beware of the job scammers. You should be highly suspicious of illegitimate companies, poorly-written job adverts, dodgy contact details and emails, unrealistic salaries, job offers without an interview and being asked for money up front. Spotting the signs of job scammers by gov.uk is licensed under Open Government Licence v3.0 (Smith and Rosser, 2021)

are over-specified as wishful employers dream up their ideal candidate.¹ Many employers will overstate their requirements in the hope they get their dream candidate. You might look at the job description and think, *I've only got 70% of what they're asking for, so I won't bother applying.* The reality is, if you've got 60% of what they are asking for, you should probably apply. It's unlikely that *anyone* will meet 100% of the job requirements.

If you see things on job adverts you don't understand or are not sure about, go and find out about them. There's a good chance it will be similar to something you already know about, or you can self-educate yourself to fill any gaps. But beware of employers over-specifying jobs.

8.4.3 Beware of unpaid internships

In the UK, it is illegal to employ people without paying them a salary. However, there are exceptions which can allow employers to take on unpaid interns depending on how they classify their employment status. See for example:

- Employment rights and pay for interns gov.uk/employment-rights-for-interns
- Targetjobs position on the law on unpaid internships: [know your rights](#)
- This article on Why I Regret Doing an Unpaid Internship (Louise, 2019)
- For more horror stories see google.com/search?q=unpaid+internships

In science, technology and engineering, unpaid internships are much less common than in other sectors as demand for skilled scientists, technologists, engineers and mathematicians (STEM) is generally high. Some employers, particularly startups, may offer company equity (such as shares) as an alternative to a salary - again you should be wary of this. Unless you're very lucky, the chances are those shares will probably be worthless. Although many startups aspire to become Unicorns, very few do. Like many people, I don't endorse unpaid internships, and I recommend you avoid them completely, see section 8.2.6. An unpaid internship might *claim* to be giving you:

- some “experience” or “exposure”
- equity-for-work
- something impressive for your CV
- opportunities to “build up your portfolio”
- improved access to paid employment at some vague point the future

Unfortunately, in the worst case, taking an unpaid internship leaves you vulnerable to exploitation by ruthless employers looking for cheap or free labour.

There is *one* exception to this: expenses paid work-shadowing for a short period of time (two or three weeks) such as spring insights described in section 7.3.

¹Using a relationship analogy, describe your ideal romantic partner. PAUSE. How many people are actually likely to have ALL of those attributes?

Otherwise, if you can't find paid employment, doing voluntary work is a much safer bet and has mental health and social benefits too, see section 7.3.

8.4.4 Beware of overselling

When people try to sell you something, you will naturally be wary of overselling and fake news, see figure 8.13.



Figure 8.13: Beware of false or misleading information in the jobs marketplace. Pool tables and free snacks are nice but are what is it *actually* like to work for a given employer? Are employers *really* as good as they say they are? Are *you* as good as you say you are? Or is it fake news? Overselling sketch by Visual Thinkery is licensed under CC-BY-ND

There's two kinds of fake news that are common in the jobs marketplace:

1. **Employers overselling themselves:** Recruitment can be a bit of a beauty contest, with employers trying to show you their best side. Some employers may make promises they can't deliver but a quick look on sites like glassdoor.com will help you evaluate employers. Even better, talk directly with actual employees of the organisation, both current and former. Is their employer really as good as they say they are?
2. **Candidates exaggerating their achievements:** It can be tempting to oversell yourself in the marketplace. An experienced reader or interviewer will be able to spot your fake news and find you out, see chapter 6 *debugging your future*

So beware of fake news and overselling. Don't believe the hype. (Ridenhour et al., 1988)

8.4.5 Beware of underselling

Likewise, you should make sure you don't undersell yourself. Know your value (financial), know your values (see chapter are 2) and try to understand how that fits with a given employer. What are the employers stated values? On the financial side, it is easy to find out about salaries, for example see:

- levels.fyi (“*Get Paid, Not Played!*”) will give you a good idea of salaries
- glassdoor.co.uk has salary information alongside employee reviews of their employers
- technical intern salaries in the UK (Wodiany, 2018)
- graduate salaries in the UK (Smith, 2021)
- The Highest Paid Internships and Placements in the UK (Louise, 2022)

Salaries in the UK for interns (and graduates) range from minimum wage to £50k and over, with everything in between. So beware of under-selling yourself, know your value, see 8.2.6. Some employers see students as a form of cheap labour that can be exploited because you're not “qualified” until you graduate. I'd think twice before working for such an employer, computing skills are in demand and there are plenty of other employers who will treat you with more respect.

Knowing your value is crucial if you're going to negotiate any job offers you receive, see section 10.4.

8.4.6 Beware of compromises

Engineering usually involves compromises and trade-offs, see figure 8.14. You will have to make some compromises in *engineering your future*. This might be in the design and implementation of your career, such as your salary, location, employer, values or something else.

To work out what compromises and trade-offs you are prepared to make, you may need to revisit the issues discussed in chapter 2.

8.4.7 Beware of the time sink

Finding employers that you are interested in and submitting high quality job applications takes lots of time. Many students under estimate the time needed to job hunt. It can be a very time consuming process for everyone, both employers and candidates alike, see figure 8.15.

Even after you have managed to:

1. Identify and articulate your skills and knowledge, see chapter 2
2. Understand what you're interested in, see chapter 2
3. Update your CV, see chapter 6
4. Consider all your options, see chapter 9
5. Target employers or sectors of interest

"The design of a general-purpose processor, in common with most engineering endeavours, requires careful consideration of many trade-offs and compromises"

Steve Furber

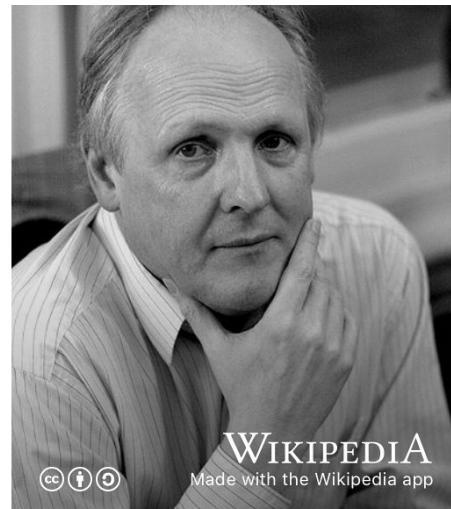


Figure 8.14: The design of a general-purpose processor, in common with most engineering endeavours, requires careful consideration of many trade-offs and compromises. (Furber, 2000) That's also true for your engineering your future too, what compromises and trade-offs are you happy to make? CC BY-SA portrait of microprocessor designer Steve Furber by Peter Howkins on Wikimedia Commons w.wiki/544E adapted using the Wikipedia app

A time sink (also timesink), time drain or time-waster is an activity that consumes a significant amount of time

Time sink



Figure 8.15: Playing the game of job hunting can be a big drain on your time and some time wasting is unfortunately inevitable. Employers will waste some of your valuable time and you'll probably waste some of theirs too. Beware of the recruitment time sink. Public domain image of a DualShock PlayStation controller by Evan Amos on Wikimedia Commons w.wiki/3VFp adapted using the Wikipedia app

.. the actual business of applying described in this chapter can be very bureaucratic. Any interviews you have will take time to prepare for (see chapter 10) and you've got loads of other calls on your time like studying and having a social life.

One way to tackle this problem is to schedule some time every week when you work on applications, see chapter 19. However, there's no getting away from the fact that finding a job can consume a significant amount of your time. So beware of the time sink.

8.4.8 Beware of rejection

For most people, rejection is a normal part of applying for jobs. Rejection can take a heavy toll on your mental health described in chapter 3 and it is often a struggle. Some employers won't even bother to reply to reject you. Welcome to the employer **black hole** described in section 6.2 and shown in figure 8.16.

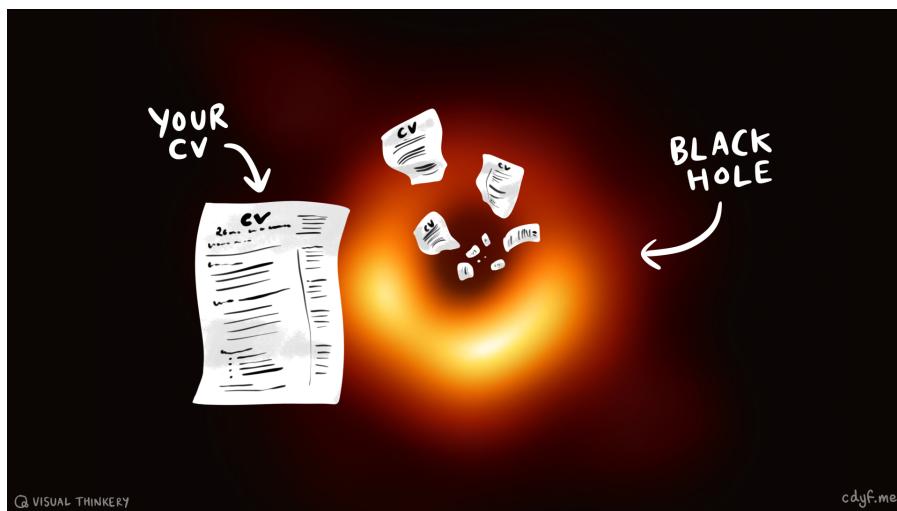


Figure 8.16: Rejection is a normal part of applying for jobs, some of your applications may disappear without trace into employers “black holes” we discussed in section 6.2. This is likely to happen with larger employers who may have a strong gravitational force on job applicants like you. This *doesn't* mean you shouldn't bother applying, but that you need to think about how to make your application stand out and avoid taking it personally if/when you don't hear back. CV black hole sketch by Visual Thinkery is licensed under CC-BY-ND, based on an original image of the supermassive black hole in Messier 87 created using the CHIRP algorithm by the Event Horizon Telescope team via Wikimedia Commons w.wiki/3RCa

If you're getting too many rejections or all your applications are sucked into black holes, it might be because you need to :

- debug your CV and job applications some more (see chapter 6)
- develop a better job search strategy (see this chapter, chapter 8)
- broaden your job search (see chapter 9)
- pay attention to your mental health, rejections can make you anxious and depressed (see chapter 3)

So beware of rejection, try not to take it personally. Beware of rejection, it's a normal part of job hunting.

8.4.9 Beware of the rollercoaster

There are highs and lows in job hunting, you will ride the job search rollercoaster shown in figure 8.17. There will be highs, you'll be invited to interviews, but there will be also be lows too, such as the inevitable rejections we discussed in the previous section 8.4.8. It will be a rollercoaster, which ends on the high of a job offer you accept. Fasten your seatbelt, enjoy the ride and good luck with your applications and interviews! May the road rise with you! (Lydon and Laswell, 1986)



WΔO

Figure 8.17: Are you ready to ride the emotional rollercoaster of job hunting? A tech project is like... sketch by Visual Thinkery is licensed under CC-BY-ND

So beware of the rollercoaster, it has ups and downs. Fasten your seatbelt.

8.5 Breakpoints

Let's pause here. Insert a breakpoint in your `code` and slowly step through it so we can examine the current values of your variables and parameters.

*** PAUSE**

- What are your current job search strategies?
- How could they be improved or tuned?
- How many jobs should you apply for?

- What are the strengths and weaknesses of using each of the parameters of `find_jobs` described above?
- Why is it important to build your network?
- How can recruiters help you?
- Why do recruiters have a bad reputation?
- How long does it take to apply for a job?
- Should I optimise for *quality* or *quantity* of job applications?
- How can you deal with the inevitable rejections that come during job hunting?

*** RESUME**

8.6 Summarising search

Too long, didn't read (TL;DR)? Here's a summary:

We've looked at search techniques that will help you find opportunities you care about. Figuring out what you want to do is tricky at times but it usually works out well in the end. Finding your future is coding your future.

This chapter is under construction because I'm using agile book development methods, see figure 8.18.

The Death Star is a fictional mobile space station and galactic superweapon featured in the Star Wars space-opera franchise

Death Star

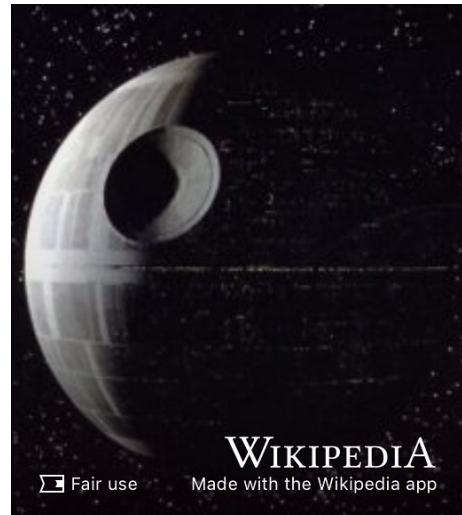


Figure 8.18: Just like the Death Star, this ~~galactic superweapon~~ book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Chapter 9

Broadening your future

Do you sometimes feel like the *weird edge case* pictured in figure 9.1? Do typical graduate destinations such as large multi-national corporations, not really make you want to *Shake Your Thang?* (Isley et al., 1988)

Perhaps you are interested in:

- using your technical skills responsibly and ethically to make the world a better place?
- starting your own business and making money for yourself, rather than other people?
- finding *hidden* or unadvertised vacancies?
- joining a startup instead of a large multinational corporation?
- venturing outside of the private sector?
- working in computing in roles beyond software engineering?

Broadening your initial job search described in chapter 8 will open up more opportunities on your horizon. This chapter will broaden those horizons and get you to think about some of the less obvious options, because I *love* weird edge cases and you should too.

Many technology jobs exist outside of technology companies, (Assay, 2020) because a lot of software is written to be *used* rather than sold. Consequently, many employers create bespoke software to fit the needs of their business. The people who build it are often employees, rather than people employed by a technology company. In the United States for example, ninety percent of IT jobs are outside the traditional tech industry. Technical jobs outside the technology sector often have the advantage of being more accessible than those within a very competitive technology sector. (Markow et al., 2019)

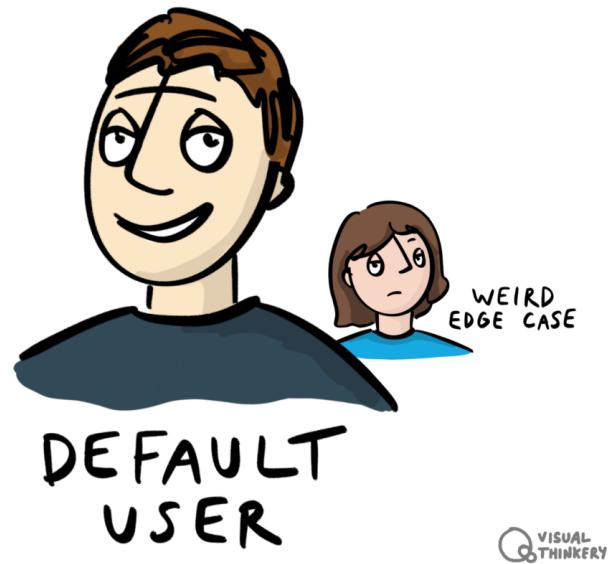


Figure 9.1: Are you a weird edge case? By default, many graduates choose a graduate scheme with big brand, often a blue-chip multinational employer. While working for these kind of employers has many benefits, they are not the whole story. This chapter looks at some of the alternatives. Default user by Visual Thinkery is licensed under CC-BY-ND

9.1 What you will learn

1. Describe the less obvious careers that computer science can lead to, besides software engineering, including:
 - Starting a business or joining a startup
 - Working outside of the technology sector
 - Working outside of the private sector (governments, non-profits etc)
 - Roles allied to software engineering that require you to be a conversational programmer
2. Recognise the social responsibility accompanying the power held by computer scientists
3. Evaluate and compare the values of an employer with your own values and ethics

9.2 Beyond software engineering

The phrase software engineering has been around since Margaret Hamilton (figure 9.2) led the development of software for the Apollo Guidance Computer in the sixties. However, the *practice* of software engineering has been around even longer right back to Ada Lovelace in the nineteenth century.

**She is one of the people
credited with coining the
term "software
engineering"**

Margaret Hamilton
(software engineer)



Figure 9.2: The role of software engineer has been around for a long time but there are plenty of other roles for computer scientists beyond software engineering. Margaret Hamilton in 1969 standing next to all of the printed code for the navigation software that she and her MIT team produced for the Apollo Guidance Computer. Public domain image via Wikimedia Commons w.wiki/3YJW adapted using the Wikipedia app

Software engineers (or software developers if you prefer) are one of the most popular roles for graduates (see e.g. figure 5.7) but there are plenty of affiliated roles that computer scientists go into besides software engineering.

- Data scientist, see prospects.ac.uk/job-profiles/data-scientist
- Database administrator (DBA), see prospects.ac.uk/job-profiles/database-administrator
- Product manager or owner, liaises with customers, management and engineers to define what a product does
- Project manager, see prospects.ac.uk/job-profiles/project-manager
- Founder (or co-founder), starting your own business (startup)
- Freelance, becoming a self-employed contractor
- Forensic computer scientist, see prospects.ac.uk/job-profiles/forensic-computer-analyst
- Business analyst see prospects.ac.uk/job-profiles/business-analyst
- Game developer (that's really just another name for software engineering) but see prospects.ac.uk/job-profiles/game-developer
- Technical writer, see section 4.5.2
- Technical sales and marketing, see prospects.ac.uk/job-profiles/technical-sales-engineer
- Test engineer (QA) see prospects.ac.uk/job-profiles/software-tester
- Research software engineer, see 9.3
- Usability engineer, often specialising in Human–Computer Interaction (HCI), User Experience (UX) or front-end development
- Security engineer, penetration testing see prospects.ac.uk/job-profiles/penetration-tester
- DevOps, sysadmin and site reliability engineering
- Patent attorney, protecting and organisations technical intellectual property (I.P.) see prospects.ac.uk/job-profiles/patent-attorney
- Consultant, see prospects.ac.uk/job-profiles/it-consultant and prospects.ac.uk/job-profiles/management-consultant

What do these roles entail?

9.3 Research software engineering

There are plenty of roles in computing working in research, either in computer science, or working alongside natural scientists, such as Physicists at home.cern or conventional scientists working at the laboratory bench. For example, there are lots of roles in research software engineering (RSE), using software engineering to facilitate better scientific research, see [society-rse.org.](https://society-rse.org/)(Woolston, 2022) For example, CERN employs ten times more engineers and technicians than research physicists, see figure 9.3. For physicists to understand the data that pours off the Large Hadron Collider (LHC), you need armies of engineers to enable the scientists to do their work. A *lot* of those engineers are working on hardware and software, and many of them won't be physicists. (Hull, 2020)

CERN employs ten times
more engineers and
technicians than
research physicists

CERN

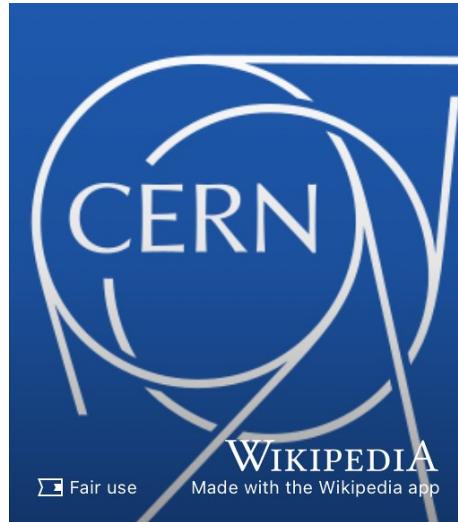


Figure 9.3: Many scientific laboratories like CERN employ lots of software and hardware engineers. Computation isn't just a fundamental part of physics, it is key to all the natural sciences so wherever you find scientists, you will also find (research) software engineers. Fair use image via Wikimedia Commons w.wiki/4qmF adapted using the Wikipedia app

An non-exhaustive list of laboratories from the UK and Europe that employ computer scientists is shown below:

- CERN, see careers.cern and careers.cern/join-us/students
- The Francis Crick Institute see e.g. crick.ac.uk/careers-study/students/sandwich-students
- The Daresbury Laboratory, see stfccareers.co.uk/students/ under *Computing*
- The Diamond Light Source diamond.ac.uk see diamond.ac.uk/Careers/Students/Year-in-Industry.html
- The European Bioinformatics Institute ebi.ac.uk see ebi.ac.uk/careers
- The Earlham Institute earlham.ac.uk e.g. earlham.ac.uk/year-industry
- The ISIS Neutron and Muon source see isis.stfc.ac.uk/Pages/Students.aspx and stfccareers.co.uk/students/ under *Computing*
- The Jodrell Bank Observatory jodrellbank.manchester.ac.uk
- The metoffice.gov.uk, see metoffice.gov.uk/about-us/careers/apprentices-graduates-and-placements
- The Plymouth Marine Laboratory pml.ac.uk see pml.kallidusrecruit.com
- The Rutherford Appleton Laboratory (RAL) see stfccareers.co.uk/students/ under *Computing*
- The Wellcome Sanger Institute sanger.ac.uk
- More like this at jobs.ac.uk

These are mainly UK opportunities, but it is a similar story around the world. Many Universities and research institutes have summer internships for computer science students working alongside researchers. For example, at the University of Manchester, summer vacancies tend to be advertised each year around April/May. Wherever you are, speak to the head of a research lab you're interested in. Ask them if they have plans to take on summer students.

If you're thinking of doing postgraduate study, see chapter 13. Commercial experience gained on a summer internship or placement year is valued by *all* employers (not just commercial ones) so doing an internship or placement during your undergraduate degree is valuable wherever you end up, see section 13.3.

9.4 With great code comes great responsibility

Computer scientists wield tremendous power in the twenty first century. We know that:

- With great power comes great responsibility (Parker, 1962)
- With great code comes great responsibility (Goldman and Schlesinger, 2018)

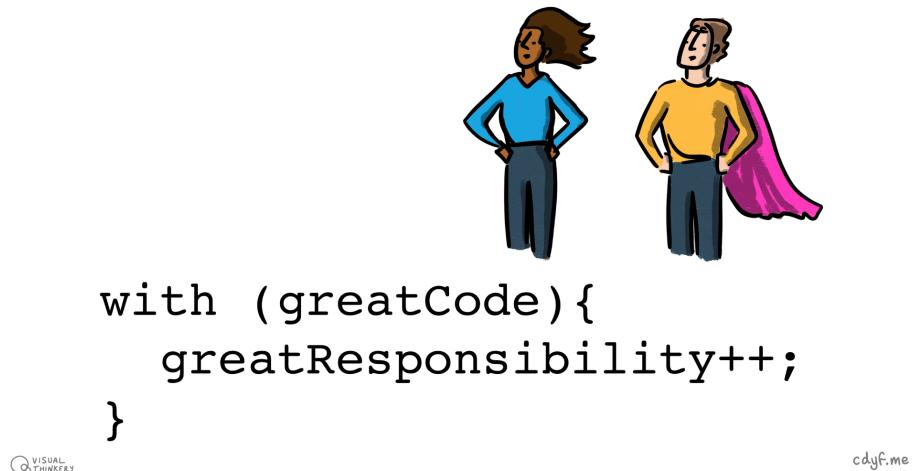


Figure 9.4: The greater your code, the greater your superpower. The greater your superpower, the greater your responsibility. What powers does computing give you and how can you use that power responsibly? (Shapiro et al., 2021) With great code sketch by Visual Thinkery is licensed under CC-BY-ND

Given the growing power of computing in the twenty-first century, computer scientists have a duty to society to use that power responsibly and justly. How can they do so? Do computer scientists need to sell their soul to the highest

bidder? How can computer science be used to make the world a better place, not just making rich people richer? Lets look at these in turn

9.5 Do you need to sell your soul?

You will sometimes hear people saying you need to sell your soul to get a job, shown in figure 9.5. See for example:

- Soul sold for less than £12 (Malham, 2002)
- Am I Selling My Soul to Work for My Company? (Bell, 2021)
- google.com/search?q=selling+your+soul+to+your+employer

The person offers their soul in exchange for diabolical favours. Those favours vary by the tale, but tend to include youth, knowledge, wealth, fame, or power.

Deal with the Devil

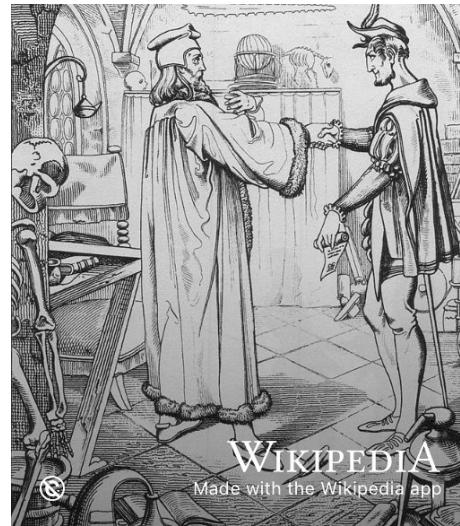


Figure 9.5: In European folklore, doing a deal with the devil is a recurring theme. Wealth, power and knowledge are some of the items that might be traded for a persons soul as part of diabolical deal. Will you need to sell your soul to the devil to get the job you want? Public domain image of an engraving by Adolf Gnauth showing Faust doing a deal with Mephistopheles on Wikimedia Commons at w.wiki/3zio adapted using the Wikipedia app

So when you're searching for jobs and researching potential employers, one of the first things you need to find out is what the values and ethical principles of an employer are, see section 8.4. This is a quick way to evaluate what makes an organisation who they are. Most employers publish their values and ethics openly, here's a small selection to give you a flavour:

- Amazon amazon.jobs/en/principles
- Microsoft microsoft.com/en-us/about/corporate-values
- Apple apple.com/compliance
- Google ai.google/principles

- Morgan Stanley morganstanley.com/about-us/morgan-stanley-core-values

With offices in more than 42 countries and more than 60,000 employees, the firm's clients include corporations, governments, institutions, and individuals

Morgan Stanley

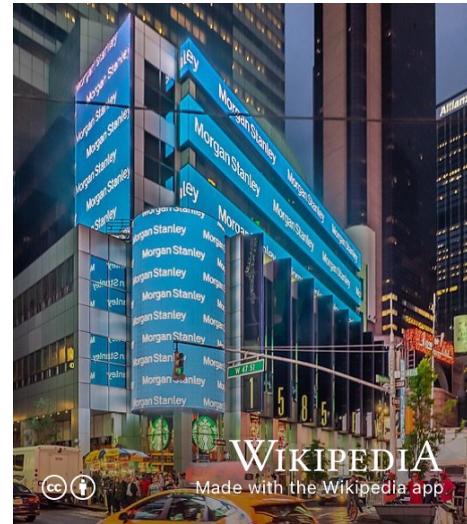


Figure 9.6: Morgan Stanley is an American multinational investment bank and financial services company headquartered in New York City. The firm's clients include corporations, governments, institutions and individuals. CC-BY picture of Morgan Stanley HQ in Times Square by Ajay Suresh on Wikimedia Commons [w.wiki/3Vnt](https://commons.wikimedia.org/w/index.php?curid=3Vnt) adapted using the Wikipedia app

Let's look at Morgan Stanley (figure 9.6) as an example, I've chosen these values because they are brief and self-explanatory. Morgan Stanley's values are to:

1. Do the right thing: act with integrity
2. Put clients first: listen to what the client is saying and needs
3. Lead with exceptional ideas: win by breaking new ground
4. Commit to Diversity and Inclusion: value individual and cultural differences
5. Give back: serve communities generously with expertise, time and money

Look at these values carefully, or choose the values of another employer you're interested in. What do they mean to you?

Do an employer's words match their actions? The words **Don't be evil** are easy to *say* but harder to *action*. Good intentions are often easier said than done.

So back to our original question, do you have to sell your soul?

- You don't have to sell your soul (Brown and Squire, 1989)
- It depends what's in your soul anyway
- If you need help doing some soul searching, see chapter 2

9.6 Computing the future

The human race faces some huge challenges in the 21st century:

- Mitigating the effects of climate change, see figure 9.7
- Tackling inequalities of wealth, income, race and gender (Stanley, 2022)
- Ensuring algorithms benefit *everyone* in society, not just the (predominantly) old rich technocratic white men that control them, see figure 9.8
- Ensuring that technology *enables* democracy, rather than undermining it, as commentators like Jamie Bartlett, Carl Miller, Rob Reich, Mehran Sahami (any many others) have argued. (Miller, 2019; Bartlett, 2018; Reich et al., 2021)
- Providing sufficient food, water, shelter, energy, education and healthcare for a growing global population which is rapidly approaching 8 billion people
- Meeting all these goals sustainably and renewably without irreversibly depleting resources

Our house is on fire. I am here to say, our house is on fire. According to the IPCC we are less than twelve years away from not being able to undo our mistakes

Greta Thunberg

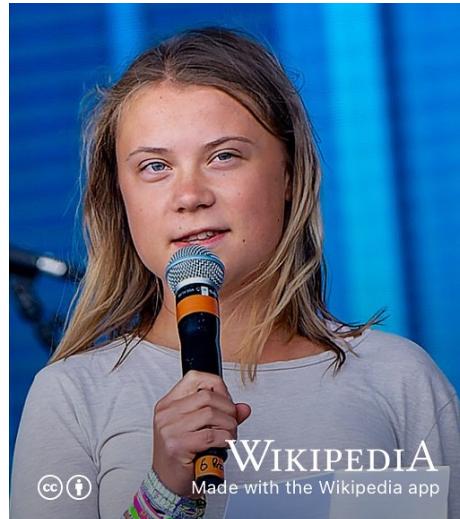


Figure 9.7: The overwhelming scientific consensus is that our climate is changing much more rapidly than we'd like. According to the Intergovernmental Panel on Climate Change (IPCC) we are less than twelve (now nine) years away from not being able to undo our mistakes. As Greta Thunberg put it, our house is on fire. (Thunberg, 2019) How can computing address this, and other global grand challenges that the human race faces in the 21st century? CC BY portrait of Greta Thunberg speaking at Glastonbury Festival of Contemporary Performing Arts in 2022 by Ralph_PH on Wikimedia Commons w.wiki/5sEK adapted using the Wikipedia App

How can computing be an ethical force for change that improves the lives people

everywhere, not just those that are lucky enough to be on the wealthier side of the digital divide?

- How can computing make a difference?
- How can **YOU** use Computer Science to make the world a better place?

Here are some examples to get you started:

- Greener computing and net zero computing, see greensoftware.foundation (Knowles et al., 2022; Anon, 2021)
- Smarter cities and more automated homes which use resources more efficiently (Hankin, 2022)
- Fairer algorithms: see algorithmic bias, responsible.ai and figure 9.8. The algorithmic equivalent of the hippocratic oath do no harm.
- More humane technology that respects users attention and minimises distraction by enabling “*Time Well Spent*”. Humane technology supports democracy, rather than undermining it, by encouraging slow thinking, rather than just fast thinking (Kahneman, 2011)
- Games for change, that have social impact rather than economic impact (profit) see gamesforchange.org
- Better education with computing:
 - as a subject in its own right: Computer Science Education
 - as educational technology that enables the teaching and learning of every subject
- Scientific computing for the benefit of humanity, see section 9.3. Creating better, cheaper and faster software and hardware for scientists and engineers, for example:
 - Improved climate modelling and weather forecasting
 - Quicker development of new vaccines and drugs, for example technologies like AlphaFold are already making a difference to drug discovery (Jumper et al., 2021)
 - Better healthcare, with electronic health records, (EHR) personal genomics and better diagnostic tools (Anon, 2022)

How will you use your superpowers of computing we mentioned in section 4.3 to make the world a better place?

9.7 Breakpoints

Let’s pause here. Insert a breakpoint in your `code` and slowly step through it so we can examine the current values of your variables and parameters.

*** PAUSE**

- How closely do a given employers values align with your own? You may need to revisit section 2.2.4.
 - You might not get a 100% match but you’re unlikely to enjoy working for an employer where your values don’t match very well at all

**She founded the
Algorithmic Justice
League, an organization
that challenges bias in
decision-making
software**

Joy Buolamwini



Figure 9.8: Are your algorithms fair or are they perpetuating biases against minority groups? Dr. Joy Buolamwini founded the algorithmic justice league to unmask harms in algorithms such as those used in facial recognition and voice recognition. CC BY-SA portrait of Dr. Joy Buolamwini by Niccolò Caranti on Wikimedia Commons w.wiki/5Jar adapted using the Wikipedia App. Thank you Dr. Joy Buolamwini for permission to use your picture.

- Are the stated values of an employer the whole story?
 - Are there any unwritten or unspoken rules?
- Is there anything missing?
- Do the employers actions match their words? What an employer *says* and *does* may be contradictory. Actions speak louder than words
- What can computing do to tackle global challenges described in section 9.6

* RESUME

Once you've thought about these questions, you stand a much better chance of working out if a given employer is a good match for you. So do you have to sell your soul as shown in figure 9.9? It depends on what you value and if an employer shares those values with you.

PRE-INTERVIEW SELF ASSESSMENT



Figure 9.9: Here's a dilemma: Do you need to sell your soul to your employer? If so, how much can you get for it? What percentage stake of your soul will they ask for and how much are you willing to give? How do your values align with those of your employer? Soul selling dialog box sketch by Visual Thinkery is licensed under CC-BY-ND

9.8 Summarising your alternatives

Too long, didn't read (TL;DR)? Here's a summary:

This chapter is under construction because I'm using agile book development methods, see figure 9.10.

The Death Star is a fictional mobile space station and galactic superweapon featured in the Star Wars space-opera franchise

Death Star

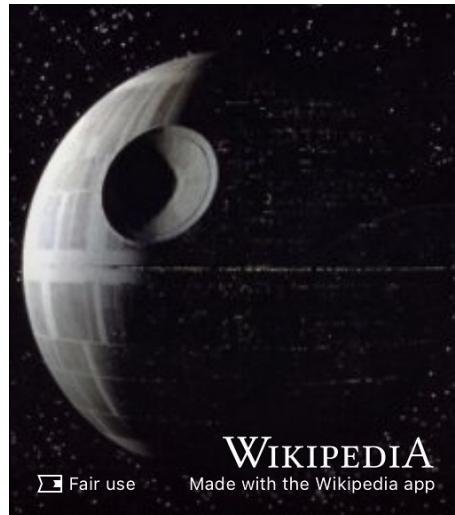


Figure 9.10: Just like the Death Star, this ~~galactic superweapon~~ book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Chapter 10

Speaking your future

Congratulations, you've been invited to an interview. It might be by telephone, video link, in person or as part of an assessment centre where you'll be asked to complete several other tasks and tests. Being invited to an interview means that your written application (CV, covering letter, application form or digital portfolio) has hit the target and is bug free, see figure 10.1. All that reading and writing has paid off. BULLSEYE!

Having passed the first stage of *reading and writing*, you move onto *listening and speaking* stage of interviewing. One of your goals is to convince the interviewers that you can articulate yourself clearly, and communicate well by listening carefully to their questions and answering with spoken natural language. These are fundamental communication skills discussed in section 4.4 on communication I/O.

If you've got an interview, you can feel good about having a bug-free CV shown in figure 10.1. Now you have a new set of problems to tackle. How can you prepare for the interview? What kinds of interviews exist and what questions might you be asked? If they offer you a job, how will you negotiate the terms, conditions and salary? Do you *really* want the job and are they the kind of people you actually want to work with everyday? If successful, you will be giving this employer:

- most of the hours of your day
- most of the days of your week
- most of the weeks of your year

This job will take a significant chunk of the next few months, year or even up to two years (and more) for a graduate scheme (Black, 2019a). It's the start of a much longer journey of up to 80,000 hours that you'll spend in employment during your life.

So you want to ensure employers are a good match and not going to waste your



Figure 10.1: If you have got an interview, then you have proved that your CV is bug free. That doesn't mean your CV is perfect, it just means that it is good enough to get you an interview with that particular employer. Congratulations! What comes next? Bug free sketch by Visual Thinkery is licensed under CC-BY-ND

time. You'll be interviewing them, as well as them interviewing you, so it will pay to have some questions prepared alongside your answers to questions you can anticipate.

10.1 What you will learn

By the end of this chapter you will be able to:

1. Identify kinds of interviews you might be invited to
2. Anticipate common interview questions, both technical and non-technical
3. Prepare questions for your interviewer by researching the employer
4. Formulate strategies for negotiating job offers
5. Calm your interview nerves

10.2 Interviews

Broadly speaking there are two basic kinds of interviews:

1. non-technical interview, sometimes called competency based interview or human resources (HR) style interview, see 10.2.2
2. technical style or coding interview, see section 10.2.3

These can be conducted in various modes:

10.2.1 Modes of interview

Interviews can be conducted in various modes:

- telephone (no visual contact)
- pre-recorded pieces to camera, you talking to your webcam
- teleconference (zoom / teams etc) with cameras and microphones turned on
- real-time face to face

10.2.2 Competency interviews

Competency interviews test some of your softer skills, find out who you are (see chapter 2), how you work in a team and if you can communicate well. A good interview strategy for any interview is to talk about what you are doing (and what you have done) as shown in figure 10.3. It might even help to have a checklist in your head of things you'd like to talk about such as:

- one work experience, see chapter 7
- a couple of projects, see section 6.6.4
- some university projects, see section 6.6.4.

If you can, make sure you've mentioned everything on your checklist by the end of the interview.



Figure 10.2: Got a video interview? Remember to toggle your mute button. Bob... by Visual Thinkery is licensed under CC-BY-ND

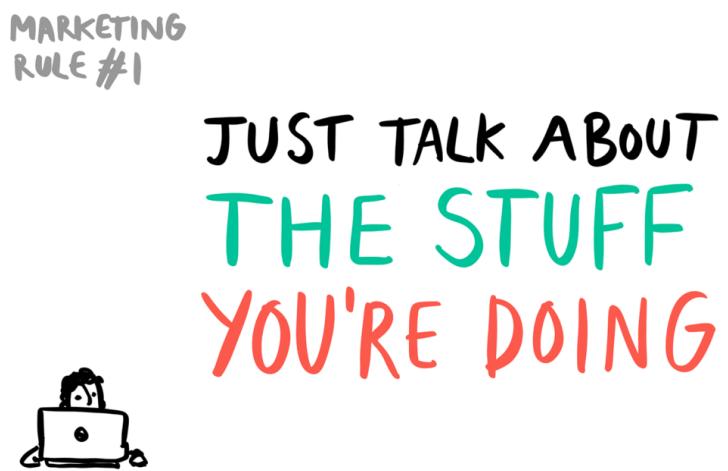


Figure 10.3: During an interview, you are marketing yourself to an employer while they market themselves to you. It can be easy to overthink your marketing. A simple marketing technique is to just talk about stuff you're doing, projects you've worked on (inside and outside University), your experience and any other relevant activities. The number 1 rule of marketing by Visual Thinkery is licensed under CC-BY-ND

Here are some common competency interview questions. Imagine you are going on a stage, prepare lines that answer these questions, rehearse them out loud in front of a mirror (or a critical friend).

1. What roles do you play in a team, with examples?
2. Tell me about a time when you showed integrity and professionalism
3. Can you give an example of a situation where you solved a problem in a creative way?
4. Tell me about a big decision you've made recently. How did you go about it?
5. Give an example of a time you resolved conflict
6. How do you maintain healthy working relationships with your team?
7. Describe a project where you had to use different leadership styles to reach your goal
8. Give me an example of a challenge you faced and tell me how you overcame it
9. How do you influence people in a situation with conflicting agendas?
10. Tell me about a time that you made a decision and then changed your mind.
11. Tell me about a time when you achieved success even when the odds were stacked against you.
12. Tell me about a time when you have provided an excellent service
13. Give me an example of a time when you have had multiple deadlines to meet? How did prioritise your workload?
14. What would you say is your main development area?

For more examples of non-technical questions see:

- google.com/search?q=competency+based+interview+questions

Since there's already tonnes of information on competency based interviews, the rest of this chapter will focus on technical interviews, also known as coding interviews.

10.2.3 Coding interviews

Many employers use technical or coding interviews to assess your harder skills. Not every employer does, but they are widely used by employers. Preparing for coding interviews is a good way to become a better engineer, so even if you don't have to face a series of really tough coding interviews, it is worth knowing about them.

Coding Caution

The competitive coding and technical interviews described in this chapter test a narrow set of algorithmic skills. While it's important to prepare for technical interviews, they have well documented limitations. (Murashenkov, 2019) Skills required for software engineering go beyond the purely technical ones. For

example, your (soft) communication skills outlined in chapter 4, and many other broader skills, are just as important as your (hard) technical abilities.

There are lots of resources to help you prepare for and practice coding interview questions, a good place to start is *Cracking the Coding Interview* by Gayle Laakmaan McDowell. (McDowell, 2015) As well as reading Gayle's book, there are lots of online resources to help you prepare for coding interviews. Before we look at those, Computer Science graduate Petia Davidova explains in figure 10.4 what she learned from failing several coding interviews at big technology companies.

Figure 10.4: Petia describes her worst software engineering interview failures. Petia demonstrates a growth mindset (section 3.7) and productive failure(s). Although she failed her earlier interviews, she learned lots from the process and went on to get a job she wanted. If you can't click on the embedded video in the figure, you can watch the full 16 minute video at youtu.be/qkeQNNjZuQk (Davidova, 2021)

Coding interviews can be tough, but preparing for them, and doing them will make you a better engineer. So if you spectacularly wipeout in your coding interview, reflect and think how can you improve next time? Perhaps you need to

- Read up on some more data structures

- Familiarise yourself with more algorithms
- Practice thinking out loud (verbally) by doing a mock technical interview?

All of these activities will help both your general professional development and your chances of success in future technical interviews. Thankfully there are plenty of resources out there for helping you get better at coding interviews, so let's have a look at some. Some of these resources are more about problem solving for recreation, rather than interview preparation, but they will all help you become a better engineer.

10.2.4 Leetcode

LeetCode is a platform to help you enhance your skills, expand your knowledge and prepare for technical interviews, see leetcode.com. There's *lots* of challenges on leetcode which can make it a daunting place for novices, thankfully neetcode.io has recorded a handy series of videos to help you navigate them all, see figure 10.5

Figure 10.5: Change your life with algorithms and data structures from neetcod.e.io. You can also watch the full 5 minute video embedded in this figure at youtu.be/TKC98Sumg2A (Code, 2022)

10.2.5 HackerRank

HackerRank allows developers to practice their coding skills, prepare for interviews and get hired. HackerRank allows users to submit applications and apply to jobs by solving company-sponsored coding challenges. Some employers use hackerrank as part of the interviewing process, so these are not just academic exercises.

Hacker Rank provide a discussion and leaderboard for every challenge, and most challenges come with an editorial that explains more about the challenge, see [hackerrank.com](https://www.hackerrank.com) and figure 10.6

HackerRank's programming challenges can be solved in a variety of programming languages (including Java, C++, PHP, Python, SQL, JavaScript) and span multiple computer science domains

HackerRank



Figure 10.6: HackerRank's programming challenges can be solved in a variety of programming languages (including Java, C++, PHP, Python, SQL, JavaScript) and span multiple computer science domains. HackerRank logo from Wikimedia Commons at w.wiki/3WAP adapted using the Wikipedia App.

10.2.6 Pramp

Pramp offers free mock technical interviewing platform for engineers. Pramp, **Practice makes perfect**, was founded in 2015 by Rafi Zikavashvili and David Glauber. As engineers, they were frustrated by the lack of resources to help them prepare for coding interviews. Find out more at pramp.com

10.2.7 Project Euler

Project Euler provides a wide range of challenges in computer science and mathematics. The challenges typically involve solving a mathematical formula or equations, see projecteuler.net and figure 10.7

Since its creation in 2001 by Colin Hughes, Project Euler has gained notability and popularity worldwide. It includes over 750 problems, with a new one added approximately every two weeks.

Project Euler



Figure 10.7: Since its creation in 2001 by Colin Hughes projecteuler.net has become internationally popular, with new problems added approximately every two weeks. (Somers, 2011) Public domain image of a painting of Leonhard Euler by Jakob Emanuel Handmann on Wikimedia Commons w.wiki/3WAV adapted using the Wikipedia app.

10.2.8 Codewars

Codewars allows you to challenge yourself on kata (), created by the community to strengthen different skills. Master your current language of choice, or expand your understanding of a new one. Find out more at codewars.com and see figure 10.8

10.2.9 ICPC

More than 50,000 students worldwide from more than 3,000 universities in 111 countries participate in over 400 on-site competitions as part of the International Collegiate Programming Contest (ICPC) see icpc.global and figure 10.9.

ICPC is organised by the Association for Computing Machinery (ACM), a global community which advances computing as a science and a profession.

There are subregional contests for ICPC, so in the UK there is the United Kingdom and Ireland Programming Competition (UKIEPC) which is part of the Northwestern Europe European Regional Contest (NWERC).

UKIEPC has been held annually since 2013 to help universities pick teams to travel to NWERC. Ask your University if they are involved, see ukiepc.info. If they aren't involved yet, you could encourage them to join. It's not just about winning, it's also about taking part.

practised in Japanese martial arts as a way to memorize and perfect the movements being executed

Kata



WIKIPEDIA

Made with the Wikipedia app

Figure 10.8: Just like martial arts, you can practice the computational arts with kata () on codewars.com and elsewhere. CC BY-SA Picture of world champion Emmanuelle Fumonde by Thierry Caro on Wikimedia Commons w.wiki/3XeE adapted using the Wikipedia app.

10.3 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

Let's imagine you're applying to work at an employer called `widget.com`. During your application you need to find out:

- What the main products and services that the organisation provides?
- Who are their clients or customers?
- Who are their biggest competitors?
- What are their values, principles and ethical policies? See section 9.7
- What sector do they principally operate in?
- Who are the market leaders in that sector?
- How is the sector changing, for example how is technology having an impact on their business?

* RESUME

an algorithmic programming contest for college students. Teams of three, representing their university, work to solve the most real-world problems, fostering collaboration, creativity, innovation, and the ability to perform under pressure.

International Collegiate Programming Contest

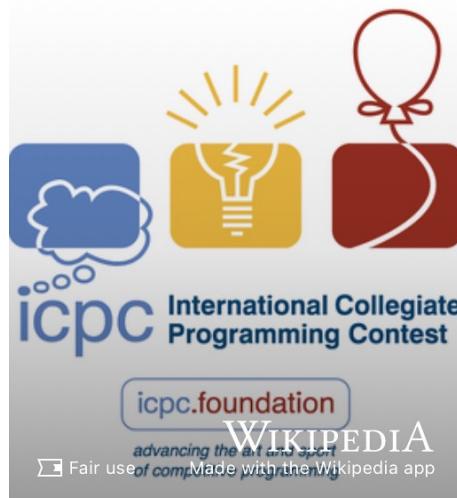


Figure 10.9: In their own words, ICPC is “an algorithmic programming contest for college students. Teams of three, representing their university, work to solve the most real-world problems, fostering collaboration, creativity, innovation, and the ability to perform under pressure. Through training and competition, teams challenge each other to raise the bar on the possible. Quite simply, it is the oldest, largest, and most prestigious programming contest in the world.” (Hacker, 2021)

10.4 Accepting job offers

If you've been successful in an interview you will receive a job offer. This is time to celebrate (congratulations!) and in some cases negotiate.

If you're not happy with the package on offer, the best time to negotiate is **before** you accept it. It is often easier to negotiate now, than get a pay rise later. This is one factor that means people leave jobs, it is often easier to get a pay rise by moving employers, rather than being promoted with an existing employer. Knowing your value is important here, its easy to find out with sites like glassdoor we mentioned in section 8.3.2. Besides the pay, you might also want to negotiate about flexible working, start date and any benefits on offer.

10.5 Communication, communication, communication!

Interviews are about a lot more than the answers you give to questions, they also test *how* you communicate, both verbally and non-verbally. We looked at communication skills in section 4.4 and throughout chapter 4 on *writing your future*. This chapter has discussed some spoken communication skills which complement your written communication skills.

**"Computing is not just
about technology – it is
actually more about
working with and for
people"**

Carole Goble



WIKIPEDIA

Made with the Wikipedia app

Figure 10.10: Computing is about much more than technology. Interviews will test some of your spoken communication skills, these are fundamental in working with and for people. (Goble, 2020) CC BY-SA portrait of scientist and engineer Carole Goble by Rob Whitrow on Wikimedia Commons w.wiki/5542 adapted using the Wikipedia app

Computing is not just about technology - it is actually more about working with and for people, see figure 10.10. So as well as testing technical knowledge, an interview (coding interview or otherwise) is a test of *how* you communicate in spoken language, not just how you speak, but how you listen and converse while you work with other people. (Goble, 2020)

10.6 Summarising interviews

Too long, didn't read (TL;DR)? Here's a summary:

We've looked at a range of platforms and competitions that can help you prepare for coding interviews. These won't just make you better at coding interviews, they'll make you a better engineer too, whatever stage you're at.

This chapter is under construction because I'm using agile book development methods, see figure 10.11.

The Death Star is a
fictional mobile space
station and galactic
superweapon featured in
the Star Wars space-
opera franchise

Death Star

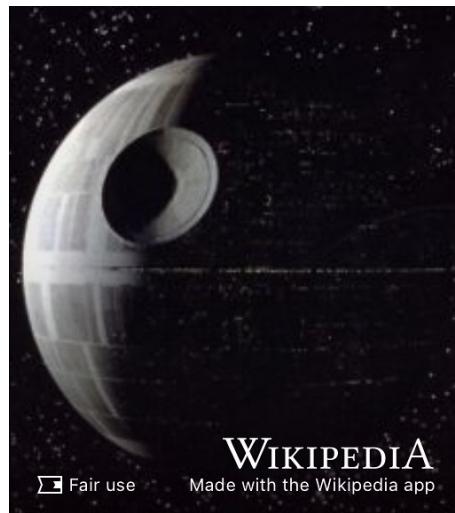


Figure 10.11: Just like the Death Star, this *galactic superweapon* book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Chapter 11

Surviving your future

Congratulations, you've just accepted an offer of employment. You nailed that interview (or interviews) and you're just about to embark on the exciting journey from the world of study to the jungle of employment. This *might* be your first SERIOUS job, so what do you have to do to survive and become a professional? What survival skills will you need, see figure 11.1. Even better, how can you thrive in your new role and take on the challenges that are coming your way? How will you avoid *diving*, ensuring you *survive* and hopefully *thrive* in your new environment?

11.1 What you will learn

At the end of this chapter you will be able to

1. Manage your manager so that you can:
 - Survive the workplace
 - Thrive in the workplace
 - Avoid diving in a workplace environment
2. Collect evidence of new workplace skills and knowledge that you develop
3. Reflect on new workplace skills and knowledge that have developed nowhere
 - ... and in the future

11.2 Survive, thrive or dive?

Starting a new job is a bit like starting a new relationship, except that it is professional rather than romantic. You've searched for and found a partner. You've been through the courtship of recruitment, this might have been quick or may have had many rounds of first and second "dates" (interviews). Once you start employment, you are both committed to each other in a serious relationship.

© VISUAL
THINKERY

cdyf.me

Figure 11.1: Will you dive, survive or thrive in your new working environment? The world of employment can be a bit of a jungle where you struggle for existence. What survival skills will you need to avoid diving (left) and how can you move beyond merely surviving (middle) towards positively thriving as a professional (right)? Jungle survival sketch by Visual Thinkery is licensed under CC-BY-ND

Simply put, there are three scenarios for you as a new employee. You'll survive, thrive or dive.

11.2.1 Survive

Your new job will go OK, you'll meet the expectations of your employer and become a valued employee. If your employer has a probationary period, you'll pass your probationary review without any problems. Most employees probably fit in to this category.

11.2.2 Thrive

Your new job will go brilliantly, you'll exceed the expectations of your employer. If you're on a fixed term contract, such as a summer internship or year long placement, they'll make you a job offer during or soon after your contract of employment expires. If you're on a more permanent contract, such as a graduate job or graduate scheme you'll be promoted and given more responsibility.

You're doing really well if you can impress your manager. Some lucky people make it into this category.

11.2.3 Dive

Your new job will go badly, you will struggle to fit in and won't meet the expectations of your employer. Once you were like star-crossed lovers, (see figure 11.2) but the relationship has turned sour and could take a disastrous dive into tragedy. (Shakespeare, 1597; Goble and Wroe, 2004)

There are several relationship problems that could lead to you breaking up with (or being dumped by) your newly estranged "lover".

- **Relationship problems:** Your relationship with your manager(s) is not going well. You've tried solving problems informally by talking to your manager but you're not satisfied with the response and want to raise a formal grievance complaint in writing. (Servant, 2021c)
- **It's not you, it's me:** You might ultimately decide to you want to hand in your notice to terminate your contract of employment and leave. (Servant, 2020b)
- **It's not me, it's you:** If things get really bad, your employer may take disciplinary action against you (Servant, 2021a) and in the worst case scenario, you'll be fired (dismissed). (Servant, 2021b)
- **Becoming unhappier:** You were looking for a job and now you've found a job. Heaven knows you're miserable now. (Marr and Morrissey, 1984) Perhaps you're not being challenged enough or given too much work that you're not interested in? Perhaps the honeymoon period is over, you've experienced hedonic adaptation and start to get disillusioned? (Santos, 2022b)

The phrase "star-crossed lovers" was coined in Shakespeare's *Romeo and Juliet*

Star-crossed

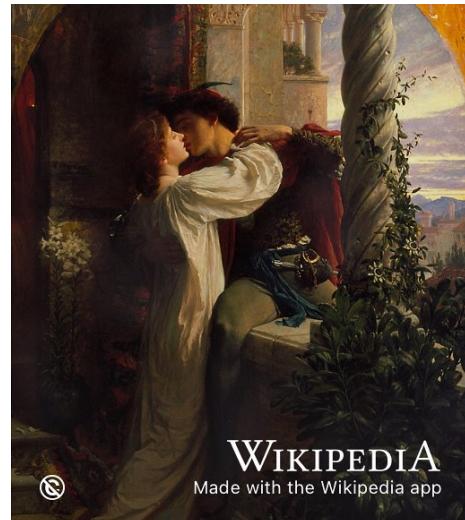


Figure 11.2: Your relationship with your employer is obviously a professional one rather than a romantic one, but that doesn't mean it can't end in tragedy like Shakespeare's star-crossed lovers in *Romeo and Juliet*. (Shakespeare, 1597) What can you do to keep your relationship with your employer healthy and happy? Public domain image of a painting of Romeo and Juliet by Frank Dicksee via Wikimedia Commons w.wiki/3DfJ adapted using the Wikipedia app

Dismissal is rare, but it **does** happen, even to interns and placement students. In this scenario in the UK, the employer has a duty to do everything they reasonably can to prevent this from happening. It's not in your employers interests to fire you because they've invested a lot of time and money in you by this point. If they have sensible recruitment procedures, those procedures will root out unsuitable candidates long before they make it to the workplace where they can cause real and lasting damage to the organisation once in post.

All employers have procedures for making sure that you can agree on work that suits both of your needs. Better employers will have better procedures to ensure this happens. Employers don't want their employees to "dive" and will try prevent this from happening wherever possible. So, speak to your manager(s) and keep the communication channels open. Talking of managers ...

11.3 Managing your manager

Building a good relationship with your manager(s) will be key to determining which of the *dive, survive or thrive* scenarios above plays out. At University, you didn't have a manager. Yes you had deadlines, but you didn't have a boss. That changes when you're an employee so it's in your interests to understand what your boss expects of you.

Software engineer Julia Evans has authored a series of programming zines, there's one called *HELP! I have manager!* (Evans, 2018) you might find useful. It will help you understand your managers job better so that you can work together more effectively. It will help you survive and thrive, not dive because it covers:

- understanding your manager's job
- setting clear expectations
- talking about problems early
- reviewing performance and getting promoted
- asking for specific feedback

The zine has the benefit of being aimed at engineers just like you. Thoroughly recommended! You might also enjoy Julia's other more technical zines such as:

- HTTP: learn your browser's language (Evans, 2021d)
- Oh Shit, Git! (Sylor-Miller and Evans, 2021)
- Hell YES! CSS (Evans, 2021a)
- How DNS works (Evans, 2021c)
- How containers work (Evans, 2021b)

11.4 Stay in school

As you develop new skills and knowledge at work, it is a good idea to collect evidence of what you've done. Whatever your career path, you'll need to keep

your CV updated. One way to think of the evidence is as *badges*, digital or otherwise. Your employer may already have training schemes that recognise and reward your accomplishments. These badges may be generic or specific to the particular sector you are working in. See chapter 12 on *Achieving your future* for more details.

11.4.1 Technical badges

Some examples of technical badges include:

- Microsoft Certifications docs.microsoft.com/en-us/learn/certifications
- Amazon Web Services Certification aws.amazon.com/certification
- Google Cloud Certification cloud.google.com/certification

Just three examples, there are many others covering both technical and non-technical skills. In many cases, your employer will encourage and possibly pay for you to get these certifications.

11.4.2 Non-technical badges

You are more than just a techie, so make sure you develop your non-technical skills as well. We introduced softer skills in chapter 4, but there's plenty of other skills to think about:

- Building resilience
- Negotiating and managing conflict
- Leadership, influence and change
- Having difficult conversations
- Emotional intelligence
- Public speaking
- Active listening

11.4.3 Online & lifelong learning

There are many online platforms for building your skills and knowledge, some examples include:

1. coursera.org
2. edx.org
3. egghead.io
4. etonx.com like the famous college it comes from, EtonX courses are not cheap but there's some good stuff here that's aimed at young people like you
5. futurelearn.com
6. khanacademy.org
7. linkedin.com/learning
8. open.edu/openlearn
9. pluralsight.com

10. skillshare.com
11. udemy.com
12. youtube.com

The choice of online learning can be bewildering. Some platforms provide free resources, others do not, but your employer may already pay for some services making them free to you while you are an employee. If that's the case, make good use of the services while you can. There's a useful comparison of four different online learning platforms here. (Chen, 2020)

11.4.4 Other training courses

Your employer may provide other courses you can go on. Again, you should make the most of these if and when they are available.

Whatever job you're doing, stay in school. Take advantage of any training on offer or go and find courses that help you develop professionally and personally. Remember that **you** are the person who cares most about your career, see section 1.5.

11.5 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

Besides collecting evidence and managing your manager, you need to manage yourself too. A proven way to do this is to periodically reflect on your work. Your employer may have procedures to help you do this, such as performance reviews or one-to-ones with your manager on a regular basis. Whatever the setup, you will benefit from taking time to reflect on:

1. What have you been doing?
2. WWW: What Went Well?
3. EBI: Even Better If?

11.5.1 What have you been doing?

Briefly describe your roles and responsibilities. What projects have you worked on? What were the main technologies that you used? As well as describing this to colleagues, you should aim to communicate this with non-specialists, people outside your field. How would you describe your job to your friends and family and terms they would understand?

11.5.2 WWW: What went well?

Are there any projects you are particularly proud of? What new knowledge or skills have you learned or improved? Remember to include both non-technical as well as technical aspects of your job. Non-technical skills include organisation, time-management, confidence, communication etc. What did you *think* about these achievements and how do they make you *feel*? What have you learned?

11.5.3 EBI: Even Better If?

What areas have you identified for improvement in the future? Again, this includes non-technical as well as technical skills. What do you *think* about these improvements and how do they make you *feel*? What have you learned and what will you do differently in the future?

11.5.4 Your managers view

If you asked your manager the same questions, would they come up with the same answers? Are there any differences between your view and your managers view of your work? If so, why do they differ?

* RESUME

11.6 Reflecting your future

The WWW/EBI activity in the breakpoint above is an example of reflective writing. You may have come across this already, but developing the ability to reflect is a key skill in:

- learning how to learn
- learning how to think more critically

The video in figure 11.3 summarises what reflection is and why its important to your future.

11.7 Summarising survival

Too long, didn't read (TL;DR)? Here's a summary:

Making the transition from education to employment can be a big and daunting change. To make the change as smooth as possible, you may need to readjust your expectations because being a student is very different to being an employee. Some key differences include:

- You have a manager when you are an employee, who you need to maintain a good working relationship with

Figure 11.3: Reflective writing puts into writing your reflective thinking in a three stage process: 1) looking back 2) analysing 3) projecting into the future. This video explains why reflection is an important *critical thinking* skill that will help you progress in the future. You can also watch the 6 minute video embedded in this figure at youtu.be/BYUy1yvjHxE. (Anon, 2014)

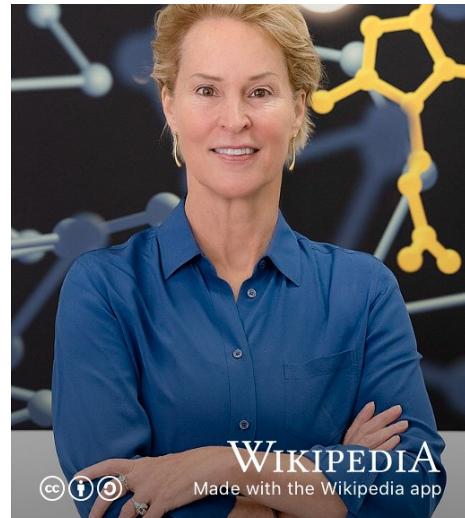
- Your performance is much more dependent on other people (your colleagues), unlike in education where you can often work more independently (e.g. when completing coursework or revising for an exam)
- You get paid, which hopefully makes it all worthwhile!

Some of the keys to *surviving your future* are therefore:

- Communicating early and often with your manager(s) about any issues you have in the workplace, see section 11.3
- Regularly reflecting on your progress, with and without your manager, see section 11.6
- Understanding that working with people is challenging for everyone (and enjoyable for many), see figure 11.4
- Staying in school. Despite leaving formal education, you should never stop learning, see section 11.4

**"Science is easy, people
are really really hard"**

Frances Arnold



WIKIPEDIA
Made with the Wikipedia app

Figure 11.4: Compared to Science, people can be really *really* hard work! (Al-Khalili, 2022) Compared to technology, engineering and mathematics, people can be really hard work too. Collaborating closely with people (your colleagues) is a key difference between your **education** and **employment**. Thankfully, people can also make your employment more enjoyable and rewarding than academic study, provided you pick the right colleagues, see section 8.2.8. Portrait of Frances Arnold in 2021 at Caltech by Christopher Michel via Wikimedia Commons w.wiki/5uJF adapted using the Wikipedia app

Chapter 12

Achieving your future

Learning is a lifelong process, a `while` loop in which you continually develop new skills and knowledge. As you loop the loop, you will collect evidence of your personal and professional achievements. Some of these achievements can be certified or “badged” in various ways. This evidence can be collected as part of your professional identity and reputation, the jar called `ME` in figure 12.1.

Evidence is a key part of your “Context, Action, Result and Evidence” (CARE) story described in section 6.7.2. So what evidence can you collect and how you can certify or badge these credentials, sometimes known as micro-credentials? (Horton, 2020)

12.1 What you will learn

After reading this chapter you will be able to:

1. Describe some the evidence and micro-credentials you can collect and badge to show your achievements:
 - during University
 - after University and throughout your professional career
2. Identify any gaps in your personal and professional achievements
3. Plan activities and set goals for future achievements that will help you to continue grow professionally and personally

12.2 Academic badges

One kind of badge you get when you finish University is your degree certificate like the one shown in figure 12.2. A degree certificate is an important offline physical (paper) badge that marks a milestone in your career. If you like gaming, its a huge `achievement unlocked` that will take you to the next

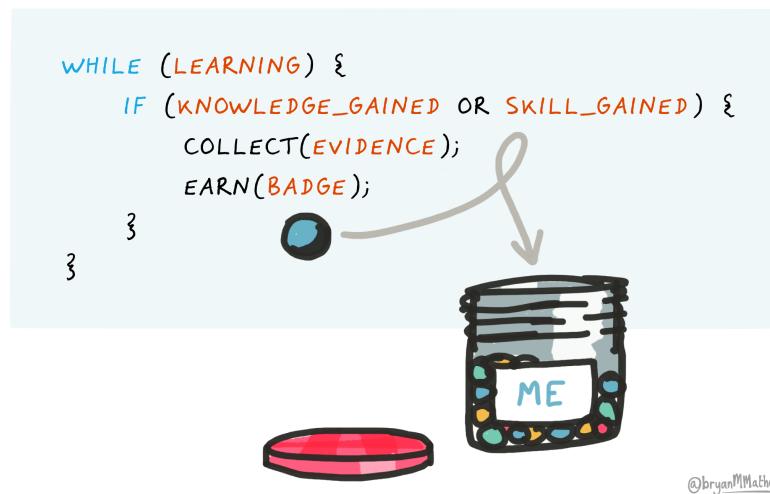


Figure 12.1: Learning is a lifelong loop where you constantly acquire knowledge and skills. You can collect evidence of your development, some of which can be certified or badged during University and throughout your professional career. Computing Badges by Visual Thinkery is licenced under CC-BY-ND

level. Your certificate also tells people that you were a member of a particular University community and that you mastered your chosen discipline to some level, Bachelors, Masters or PhD.



Figure 12.2: Level up, achievement unlocked! A degree certificate is a milestone that provides evidence of your academic knowledge and skills gained while at University. A certificate is just an offline badge by Visual Thinkery is licensed under CC-BY-ND via Doug Belshaw

Degree certificates are an important badge, but they don't give very many details of your professional and personal story while at University. You could give more details by providing:

- your overall degree classification: first, second, third etc
- your individual module grades, for example in an academic transcript or by listing them on your CV, see section 6.6.2
- your projects, see section 6.6.4
- your portfolio of work, if you have one

This data give a *bit* more detail than a degree certificate does not but it is limited to purely academic achievements. You are *much* more than your grades, because there's a lot about your character that is difficult or impossible to measure, see figure 12.3.

Employers will often want to see more detailed evidence of your character and your softer skills than those provided by degree certificates and grades. While academic achievements paint some broad brushstrokes of your professional identity shown in figure 12.4, they don't help employers see the finer details or much



Figure 12.3: Your grades give more detail than a degree certificate but they still don't say much about you. I am more than just my grades sketch by Visual Thinkery is licensed under CC-BY-ND

of the evidence.

12.3 Digital badges

Digital badges provide a solution to this problem, just like your degree certificate is a verified badge of your achievements, a digital badge does the same but in a digital way. Rather than being physical, a digital badge is virtual and transferable. It's just a *.png graphic file which has been digitally signed and contains metadata. This means it can be displayed on your CV, on social media such as LinkedIn or "stacked" into a digital portfolio collection as shown in figure 12.5.

The Mozilla Foundation have defined an open standard for digital badges (Anon, 2019b) so that badges can be issued, hosted and certified by different organisations. (Belshaw, 2019)

A digital badge has certified data locked inside (shown in figure 12.6) which details the achievement it has been awarded for including:

- **recipient** that's you!
- **issuer** the organisation awarding the badge, e.g. Poppleton University
- **badge name** e.g. PASS leader badge (see figure 12.7)
- **badge image** e.g. a digital logo



Figure 12.4: What kind of picture would you like to paint of your professional identity? Open Badges paint a better picture... by Visual Thinkery is licenced under CC-BY-ND

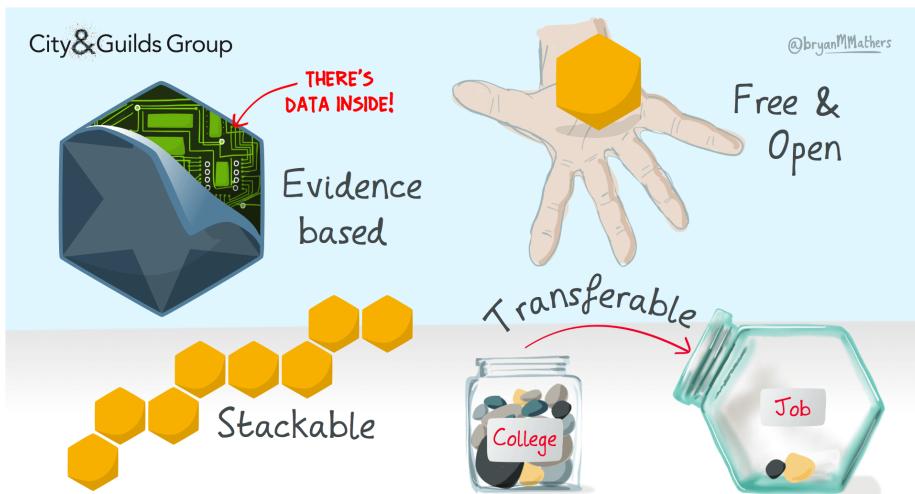


Figure 12.5: Digital badges have certified evidence inside, can be transferred between education and employment and can be stacked into collections or portfolios. Properties of Open Badges by Visual Thinkery is licenced under CC-BY-ND for the City and Guilds of London Institute

- **evidence URL** a link to evidence

All this information is coded so that only the recipient and issuer can manipulate it, for example by associating an email address with it.

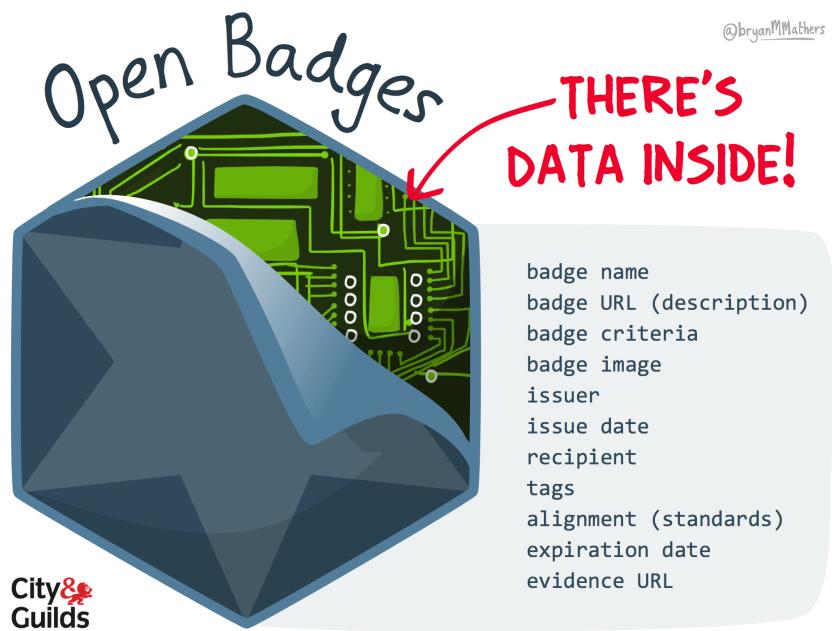


Figure 12.6: Open digital badges have data, like the **issuer** and **recipient**, locked inside them so they can be verified. There's data inside open badges by Visual Thinkery is licenced under CC-BY-ND for the City and Guilds of London Institute

Anyone such as your University or employer can issue badges, so for example, the University of Manchester issues badges for leaders of its Peer Assisted Study Scheme (PASS) www.peersupport.manchester.ac.uk. An example of a PASS leader badge is shown in figure 12.7.

Badges can be used for a wide range of different kinds of achievements shown in figure 12.8.

1. **Membership** being a member of an organisation or group
2. **Participation** participating in communities
3. **Capability** demonstrating capability with some skill
4. **Mastery** mastering a set of skills



Figure 12.7: An example of a digital badge awarded to Peer Assisted Study Scheme (PASS) leaders at the University of Manchester who have mentored and helped others students during their academic study. If you're a University of Manchester student, you can see other badges available at wiki.cs.manchester.ac.uk/index.php/Badges (UoM login required)



Figure 12.8: There are many different achievements which badges can be awarded for including membership, participation, capability and mastery. Badge taxonomy by Visual Thinkery is licensed under CC-BY-ND

12.4 Other digital badges

There are other digital badges for evidencing your achievements besides the open ones described in this chapter. Like open badges, they also provide certifiable evidence of professional and personal achievements, see figure 12.9:

- Amazon Web Services badges credly.com/organizations/amazon-web-services/badges
- Certificates from Microsoft and Google, see section 11.4.1
- Certificates from [edx.org cs50.harvard.edu/x/2020/certificate](https://edx.org/cs50.harvard.edu/x/2020/certificate)
- Certificates from coursera.org/professional-certificates
- Certificates from redhat redhat.com/en/services/certifications etc



Figure 12.9: An example of an Amazon Web Services badge awarded by credly.com for an AWS certified developer.

We have focussed on technical achievements here, but there are non-technical achievements too.

12.5 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

*** PAUSE**

- Besides the badges and certifications already discussed, what others exist?
- For the skills and knowledge you already have, what evidence do you have for it?
- Where are the gaps in own skills or knowledge?
- What evidence can you collect that you are developing these skills and knowledge?
- What parts of this evidence are you able to badge and certify?

*** RESUME**

12.6 Summarising your achievements

Too long, didn't read (TL;DR)? Here's a summary:

Open digital badges let you take your achievements from the many places you learn, and combine them into a portable portfolio that tells a digital story about your learning. The data inside a badge is shown in figure 12.10 and gives an employer a more detailed and evidenced view of your professional development than traditional physical badges like degree certificates.

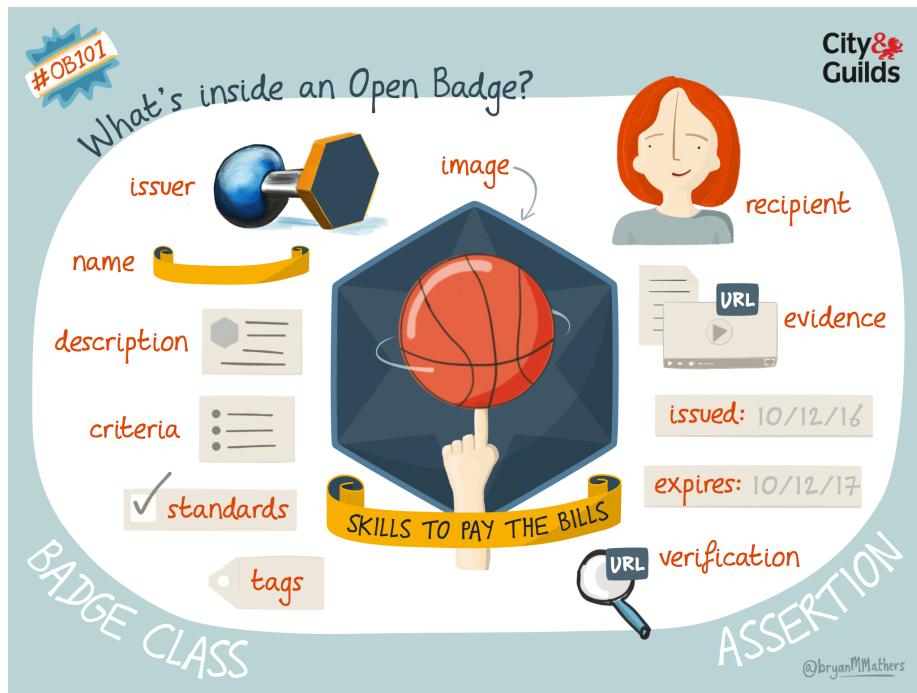


Figure 12.10: Open digital badges have certifiable metadata locked inside. What's inside an open badge? by Visual Thinkery is licenced under CC-BY-ND for the City and Guilds of London Institute

Chapter 13

Researching your future

So you want some more, eh? Your undergraduate degree has whetted your appetite. What are the options for postgraduate study and research? Where can they take you and will further study be worth investing your time and money in? You are a curious character. You like the idea of pushing the boundaries of human knowledge a little further, maybe you even fancy yourself as a modern day Ada Lovelace or Alan Turing?

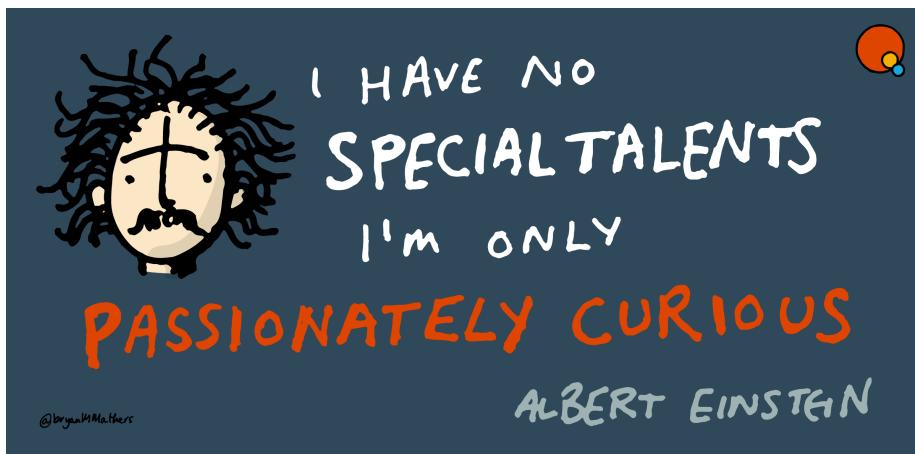


Figure 13.1: I have no talents, I'm only passionately curious. Are you passionately curious? Is further study or research the right path for you? Are you the next Einstein or Einsteiness? This chapter looks at some of the possibilities for research and further study. Curiosity by Visual Thinkery is licensed under CC-BY-ND

13.1 What you will learn

At the end of this chapter you will be able to:

1. Describe the costs of postgraduate study and research
2. Discuss the benefits of postgraduate study and research

13.2 Where to start

A good place to start if you're looking for a masters or PhD are:

- Apply directly to Universities for postgraduate study, if there is a specific group or course you are interested in. See also:
- findamasters.com for postgraduate study, a directory of Masters degrees and postgraduate qualifications at universities around the world
- findaphd.com for postgraduate research, a large database of PhD opportunities
- jobs.ac.uk/phd also lists PhD opportunities, not just in the UK, see jobs.ac.uk/phd

13.3 University or the Real World?

When you graduate, you might think you need to choose between University or the “real world”, see figure 13.2. That’s a popular misconception and a false dichotomy. In computing it is common for people to cross back and forth several times between industry (the so-called real world) and academia (the ivory towers).

Commercial or real-world experience is valuable in academia and academic experience is valuable beyond academia. So, if you think you need to choose between the two, think again. Doing an internship or placement with a commercial company can be useful background for a career in academia, both in research and teaching. Lots of research is done in partnerships between commercial companies and Universities, some of these create startups, University spin-offs or corporate spin-offs etc.

Your learning shouldn’t stop when you finish University either, so you might work for a bit, study for a bit, work a bit more and so on. Some of this learning might be formal and academic with qualifications, others might be badged in different ways. This is called lifelong learning, see section 14.8 and chapter 12.

So will you choose University **OR** real life in the longer term? With all the squiggles and zigzags, your career is likely to be a bit more complicated than that... [Tupper and Ellis (2020);]



Figure 13.2: Which path will you choose, University or the real world? Perhaps you can have both? How many times can you zigzag between the two? Real world sketch by Visual Thinkery is licensed under CC-BY-ND

13.4 Ask beyond academia

If you're thinking about doing a PhD and going into research, it's obviously a good idea to talk to Professors and academics about what is involved. Don't just ask Professors though, talk to PhD students, postdocs and especially people who did a PhD but didn't go into academia. Most people with a PhD don't go on into academia as a career. (Ruben, 2017) This means that asking a Professor if you should do a PhD is a bit like asking a Hollywood film star if you should become an actor. The film star can only give you part of the story that is likely to suffer from survivorship bias. (McCormack, 2017)

13.5 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

1. When is the best time to do a masters, straight after your undergraduate degree or after working for a while?
2. How much does a Masters degree improve career prospects?
3. How much does a PhD improve career prospects?
4. Is postgraduate study and research really worth all the pain and suffering?
5. What careers can a PhD lead to?

*** RESUME**

13.6 Signposts from here on research

A good place to start if you're thinking about doing a PhD (or trying to get through one) is *How to get your PhD: A Handbook for the Journey* by Gavin Brown. (Brown, 2021a)

I wish I'd had this book when I was a PhD student! I'm not just saying that because Gavin is a colleague of mine but this is a genuinely useful book which quickly tackles a wide range of issues you'll encounter during a PhD from the technical to the psychological. The second half also contains a range of short viewpoints on doing a PhD from people including Nancy Rothwell, Victoria Burns, Steve Furber, Lucy Kissick, Hiranya Peiris, Melanie Leng, Jeremy Wyatt, David Hand, Carolyn Virca, Shakir Mohamed, Jonny Brooks-Bartlett and Jennifer Polk. If you're serious about doing a PhD, you should read Gavin's guidebook.

13.7 Summarising further study and research

Too long, didn't read (TL;DR)? Here's a summary:

This chapter is under construction because I'm using agile book development methods, see figure 13.3.

The Death Star is a fictional mobile space station and galactic superweapon featured in the Star Wars space-opera franchise

Death Star

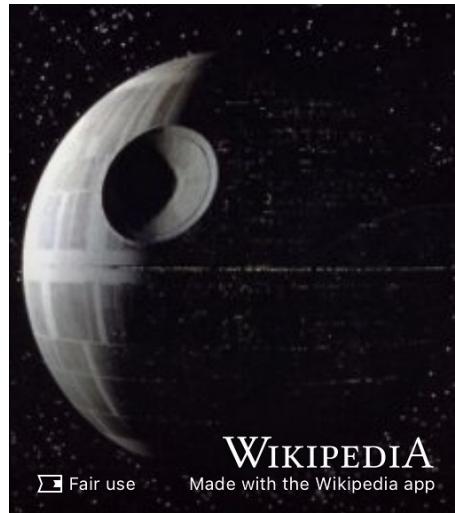


Figure 13.3: Just like the Death Star, this ~~galactic superweapon~~ book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Part III

DEPLOYING

Chapter 14

Ruling your future

In 2005, the scientist and engineer Phil Bourne starting publishing a series of articles which distilled people's hard won knowledge into *Ten Simple Rules*. (Bourne, 2005) Over a decade more than 1000 rules were published in over 100 articles in the scientific journal *PLOS Computational Biology*. (Bourne et al., 2018) These articles offer a huge range of advice from making the most of a summer internship (Aicher et al., 2017) to teaching programming (Brown and Wilson, 2018) and even winning a Nobel Prize. (Roberts, 2015) Articles as lists, or "listicles" as they are sometimes known, are a convenient way to summarise key points. So here are *Ten Simple Rules for Coding Your Future*: the too long, didn't read (TL;DR) summary of this guidebook.

14.1 Know who you are

There is a lot more to you than your degree. There is a lot more to you than your grades. Yes, you've spent (or will be spending) three or four years getting your degree. Use this time to identify your weaknesses and work out how to improve them. Knowing your future depends on knowing who you are now, see chapter 2 and figure 14.2.

Being able to articulate your self-awareness to employers will make writing a CV (chapter 6) and interviewing (chapter 10) a lot easier. Both rely on you be able to express yourself in written and spoken language. You should aim to strike a balance between humbly underselling yourself (see section 8.4.5) and proudly overselling yourself, see section 8.4.4. If you have digital profiles like those described in section 8.3.3, how accurate a reflection of you are they? Is your digital alter ego at odds with reality?



Figure 14.1: Ten Simple rules for coding your future. Know who you are, look after yourself, use what you have, grow your networks, always make new mistakes, help and thank, look beyond the obvious, stay in school, step outside your comfort zone and (*most importantly*) don't give up! Figure by Visual Thinkery is licensed under CC-BY-ND



Figure 14.2: How well do *really* you know yourself? Know who you are sketch by Visual Thinkery is licensed under CC-BY-ND

14.2 Look after yourself

Studying at University can be enjoyable but it can also make you stressed, anxious and depressed. If you neglect to look after yourself mentally and physically, things can start to fall apart. Choose your reference points carefully, try not to compare yourself to the person at the top of the class: see Carmen's advice in chapter 24. Ask yourself, am I doing better than last time? Be kind to yourself because nurturing yourself now will nurture your future, see chapter 3.



Figure 14.3: It's important not to neglect your body, mind and soul when you're working hard. Look after yourself by Visual Thinkery is licensed under CC-BY-ND

At University, you are quite likely to be surrounded by people who have more experience, better knowledge, higher grades or more skills than you. That might not have been the case at high school if you were top (or near the top) of your class. While it is natural to compare yourself to the best, you should do so with caution. The important thing is that you are working towards the best version of yourself, not necessarily the best in your class anymore.

14.3 Use what you have

It's too easy to fall into a trap of thinking *if I had...* when job hunting. Use what you already have, see figure 14.4.

This rule is borrowed from software engineer Greg Wilson @gwilson in figure 14.5, who probably adapted it from a quote frequently misattributed to Theodore Roosevelt (Brewton, 2014).

Getting a job offer is competitive and can be cut-throat. You might find yourself falling into poor habits of mind:



Figure 14.4: Use whatever resources you have at your disposal rather than thinking about the resources you don't have. Use what you have by Visual Thinkery is licensed under CC-BY-ND

**"Start where you are,
use what you have, help
who you can"**

The Carpentries



Figure 14.5: "Start where you are, use what you have, help who you can." — Greg Wilson at third-bit.com and software-carpentry.org. CC BY Portrait of Greg Wilson at The Carpentries via Wikimedia Commons w.wiki/3a6V adapted using the Wikipedia App.

- “*If I had a better degree from a different university, I’d be more successful...*” see section 1.6
- “*If I had more experience, more voluntary work, more internships, I’d stand a better chance...*” see section 7.3
- “*If I’d done more projects and extra-curricular activities...*” etc see section 6.6.4
- “*If I’d got better grades at school and Uni...*” see section 12.2
- “*If only I’d worked harder...*” see 2.4.1
- “*If I was more confident at speaking and interviews ...*” see chapter 10 on *Speaking your future*
- “*If I’d been to a different school, I could be more successful...*” see the 93percent.club (Nye, 2021; Verkaik, 2021)

Coulda. Woulda. Shoulda. This is all the usual dialogue you can expect from your inner critic. Acknowledge these thoughts, see section 3.4, then try distance yourself from them. Start from where you are, use whatever you have and help who you can.

14.4 Grow your networks

Grow your networks, make use of all the contacts you have and foster new connections where you can. Improve the connections you already have by spending time talking to people and hearing what they have to say. People can help you, *especially* those you’re not particularly close to, see figure 14.6

Remember that the weaker ties in your network (see section 8.2.5) may be more important than your stronger ties, especially when it comes to finding jobs. It’s not (just) what you know, but *who* you know.

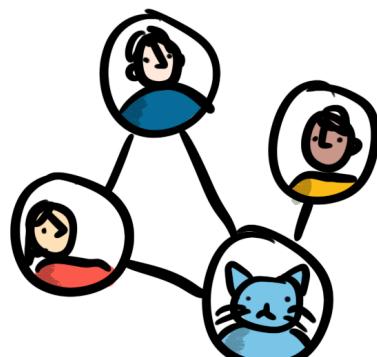
14.5 Always make new mistakes

You can classify your mistakes and failures into two categories:

1. Productive mistakes: those you learnt from
2. Unproductive mistakes: those you didn’t learn anything from (and risk repeating)

Mistakes and failure are inevitable in life, but productive mistakes are going to help you much more than unproductive ones (Petroski, 1992). That doesn’t just mean you should “*fail fast, fail often*” (Babineaux, 2013) or “*move fast and break things*”, but to consciously learn from any mistakes you make so that you don’t repeat them. One way to turn unproductive mistakes into productive ones is deliberately and consciously reflect on why you made them. This is part of the growth mindset we discussed in chapter 3.

In a growth mindset, mistakes can be good, but the fear of making them is not. You are more likely to take more chances when you’re unafraid to fail, and this



**GROW YOUR
NETWORKS**

Q VISUAL
THINKERY

cdyf.me

Figure 14.6: Grow and use your networks, both the strong ties and the weak ties described in section 8.2.5. Grow your network by Visual Thinkery is licensed under CC-BY-ND

will improve your chances of success.

Many education systems around the world don't teach people how to fail, because they put too much emphasis on success (as measured by grades and rewards) rather than progress, learning and happiness. (Lahey, 2016; Kohn, 2018) So as the angel investor Esther Dyson once said, "Always make new mistakes", see figure 14.7

**"Always make new
mistakes"**

Esther Dyson



Figure 14.7: Mistakes are inevitable in life, so there's no shame in making them especially if they are new. Making *new* mistakes can be a form of productive failure that you learn from rather than a source of unproductive failure that you repeat (old mistakes). Portrait of Esther Dyson by Christopher Michel (CC BY-SA) via Wikimedia commons w.wiki/3TEY adapted using the Wikipedia App.

So:

- If you've got some harsh feedback on your CV, how can you make less buggy in the future?
- If you've applied to lots of companies and not even had a reply yet, how you improve your job search strategy?
- If you've neglected to develop interests and projects outside of work, how can you rebalance?
- If you crashed and burned in an interview, how can you use the experience to do better next time?
- If you failed to get the promotion you thought you deserved, what will you do differently in the future

14.6 Help and thank who you can

There are good reasons to be grateful, showing gratitude doesn't just help other people, it helps you too see figure 14.8



Figure 14.8: Help and thank who you can. Help by Visual Thinkery is licensed under CC-BY-ND

Join a team by helping someone, be a team player, help others, thank others for their help.

14.7 Look beyond the obvious

Be flexible in approach. Don't just target big employers that you've heard of, there are plenty of startups and smaller organisations you've never heard of who have lots to offer, see figure 14.9

It's not just London (see chapter 16), and other big cities. Look beyond graduate schemes, look beyond graduate jobs. Broaden your horizons and your job search, see chapter 9.

You are not just a techie, Either.

© VISUAL
THINKERY

cdyf.me

Figure 14.9: Look beyond the obvious, don't restrict your job search to employers everyone has heard of as there are many more opportunities on offer. Binoculars by Visual Thinkery is licensed under CC-BY-ND

14.8 Stay in school

Part of what you learn during your education is *how* to learn. But your learning shouldn't finish when you leave University, see figure 14.10 chapter 12

Computer science is a young and rapidly changing discipline which means you can not afford to be left behind. Don't let your mind retire. (Haim et al., 2012) Never stop learning, see chapter 12.

14.9 Step outside your comfort zone

Research has shown that we often learn more when we are uncomfortable. You will often learn more when you step outside your comfort zone and reach for things that push your limits. Sometimes when you think you are learning. Learning is an active process that requires you to do things rather than just being a passive observer or consumer. (Deslauriers et al., 2019; Barshay, 2022)

Am I being insensitive asking people to step outside their comfort zone when we've all been stretched beyond breaking point during COVID-19, climate change and global economic turmoil? We're all going to need to continue to step outside of our respective comfort zones in order to meet the challenges we face around the world, see figure 14.11.

This takes courage, but that's often when you learn most. So step outside your comfort zone if you're feeling brave enough to learn.



Figure 14.10: Stay in school because learning is a lifelong process, a `while` loop in which you continuously develop new skills and knowledge. Stay in school sketch by Visual Thinkery is licensed under CC-BY-ND



Figure 14.11: You learn and grow more when you step outside your comfort zone. Comfort zone sketch by Visual Thinkery is licensed under CC-BY-ND

14.10 Don't give up

Job hunting can be hard. Job hunting can be stressful. Job hunting can be time consuming. Some employers will waste your valuable time, see section 8.4.7. Some employers will reject you but try not to take it personally, see 8.4.8. Job hunting may affect your mental health, see chapter 3. The important thing is to not give up, see figure 14.12. Try to make any failure productive, rather than unproductive, see section 14.5.

"Don't Give Up"

Don't Give Up (Peter Gabriel and Kate Bush so...

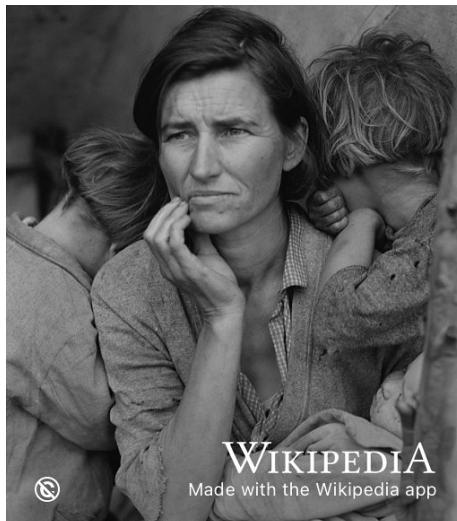


Figure 14.12: You were taught to fight, taught to win, perhaps you never thought you could fail? Don't give up, because you have friends. Don't give up, you're not beaten yet. Don't give up, I know you can make it good. (Gabriel and Bush, 1986) Public domain portrait of a *Migrant Mother* by Dorothea Lange via Wikimedia Commons w.wiki/3cRg which inspired the song Don't Give Up by Peter Gabriel and Kate Bush. (Gabriel and Bush, 1986)

14.11 Ten simple summaries

This chapter is under construction because I'm using agile book development methods, see figure 14.13.

The Death Star is a fictional mobile space station and galactic superweapon featured in the Star Wars space-opera franchise

Death Star

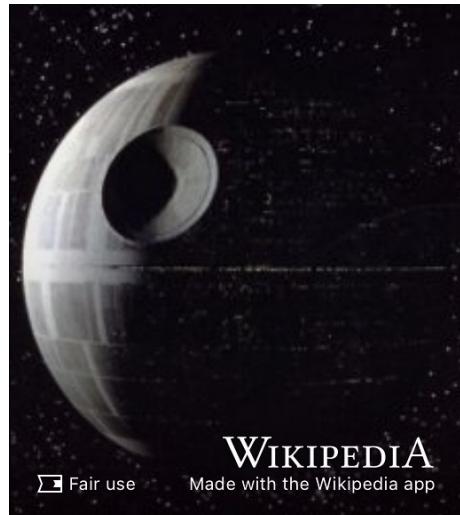


Figure 14.13: Just like the Death Star, this ~~galactic superweapon~~ book is under construction. As of 03 December, 2022 this book is an estimated **37% complete**. Image of agile weapon engineering in *Star Wars* via Wikimedia Commons w.wiki/5N6q adapted using the Wikipedia app

Chapter 15

Hacking your future

It's very easy to overlook mistakes in your own writing because it's difficult to be both an *author* and an *editor* of the same text. That's true of any written communication such as a covering letter, personal statement, email, essay or any message that you write. Mistakes are particularly common in CVs (or résumés) because they're so personal. You can spend *hours* carefully polishing the words and the formatting but not see a fatal error at the top of page one. Hacking¹ other people's CVs will help you debug and improve your own. You may need to use ingenious hacks like the temporary fix to the bridge in figure 15.1. Fixing other peoples bridges will help you improve your own bridges, see section 0.5.1. You'll build better bridges to more interesting and ambitious destinations.

15.1 Hack their CVs

A useful hacking technique is to take somebody else's stuff and fix or improve it. That works for written communication as well as actual code. The dogfooding technique described in section 4.5.1 is a useful hack which you can use by:

1. Eating your *own* dogfood by reading your own written work ALOUD
2. Eating *somebody else's* dogfood by:
 - Hacking a friend's or peer's CV by swapping with them and giving them constructive feedback
 - Hacking the fictitious CVs below by ranking them against a job advert in section 15.4. Who would you want to interview and why?
3. Eating your own dogfood again, re-reading and re-editing repeatedly
4. Persuade someone else to eat your dogfood - get feedback from as many people (and bots see section 6.7.6) as you can

¹The term hacking is a horribly overloaded word with many different meanings, but I'm using it here to mean an "ingenious temporary solution to a problem"



Figure 15.1: Your CV is a bridge which enables you to cross from where you are now to where you'd like to be in the future. Like the bridge in this picture, the CV's in this chapter are all faulty in some way, can you fix them? You may need to use ingenious hacks and kludges like the one shown here on the Million Dollar Bridge in Alaska . Public domain image of kludgy repairs adapted from an original by Jet Lowe on Wikimedia Commons w.wiki/3Uvn

So, here are some fictitious CVs for you to hack, from students of Computer Science. They are based on CVs I've seen, warts and all, with personal information removed and anonymised. Can you spot their triumphs and tragedies? Can fix their CVs and work out which candidate is best for the sample job description at CoolTech in section 15.4? Can you hack their future?

Special thanks to Toby Howard and Sean Bechhofer for coming up with some of these silly names for fictional students in computing. Please direct any complaints about the terribly geeky puns to Toby and Sean! Can you spot all the bad jokes? Thanks also Ben Carter and Penny Gordon Lanes in the careers service at the University of Manchester, some of these CVs are based on examples they have collected and anonymised.

15.2 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

When you read these CVs make a note of:

1. **What Went Well?** (WWW) What do you like about any given CV, what have they done well?
2. **Even Better If?** (EBI) What could be fixed or improved, can you hack it?
3. **Their Rank order** (1,2,3...) Who is top of your list to interview? Who is going in the bin and why?

* RESUME

Imagine the person is real, what would you tell them about their CV if they'd given it to you for advice without hurting their feelings? How could you be a critical friend by giving them actionable feedback?

15.3 Sample CVs

Hacking other people's CVs will help you improve your own because you're putting yourself in the shoes of your reader. Here are some samples:

15.3.1 Penelope Tester

Penny Tester, or Pen as her friends call her, loves cybersecurity and reverse engineering. She has a real passion for finding vulnerabilities in software and hardware. Just don't call her a hacker she dislikes that word, see 15.2.



Figure 15.2: Penny Tester's full CV can be viewed at cdyf.me/Penelope_Tester.pdf

15.3.2 Rick Urshion

Rick is a big fan of functional programming and loves solving problems with languages like Lisp, Haskell, Clojure, Erlang and Scala. He really doesn't like side-effects but tries to avoid getting into a state about it. His critics say he can be inefficient but Rick insists he's just lazy, see figure 15.3.

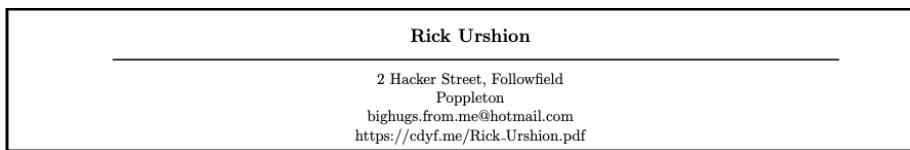


Figure 15.3: Rick Urshion's full CV can be viewed at cdyf.me/Rick_Urshion.pdf

15.3.3 Marge Conflict

Marjorie loves version control, she has very high emotional intelligence (EQ). This superpower enables her to recognise emotions in herself and others so that she can resolve people's differences, see figure 15.4.



Figure 15.4: Marge Conflict's full CV can be viewed at cdyf.me/Marge_Conflict.pdf

15.3.4 Michael Rokernel

Mike lives in Los Angeles and loves operating systems, but not if they get too bloated, see figure 15.5.

Résumé: Mike ROKERNEL

Figure 15.5: Mike Rokernel's full CV can be viewed at cdyf.me/Mike_Rokernel.pdf

15.3.5 Florence Ting-Point

Flo loves maths and is a particularly big fan of floating-point arithmetic, see figure 15.6.

Florence Ting-Point

COMPUTER SCIENCE STUDENT
Flat 5, The Quayside, Salford, Poppletonia
(+82) 10-9030-1843 | flo.ting.point@gmail.com | cdyf.me/Flo_Ting-Point.pdf | floatingpoint.flo.ting-point.com | floatingpoint.flo.ting-point.com

Figure 15.6: Flo Ting-Point's full CV can be viewed at cdyf.me/Flo_Ting-Point.pdf

15.3.6 Peter Byte

Peter and his twin sister Peta, both love big data, machine learning, statistics, data science and Artificial Intelligence (AI). They come from a big family with nine siblings, Deca, Hector, Kilo, Megan, Giga, Terry, Exa, Zita and Yotta. They are wildly ambitious, but critics say the Byte family have been terribly over-hyped, see figure 15.7.

CV – PETER BYTE		
Home Address: Address	7 Overflow Lane 3 Whitehill	Term Rushinghome

Figure 15.7: Peter Byte’s full CV can be viewed at [cdyf.me/Peter_BYTE.pdf](http://cdyf.me/Peter_Byte.pdf)

15.3.7 Polina Morphism

Polly *loves* object-oriented programming. She has lots of siblings, and a cousin called Isa. Instead of a CV or résumé, Polly has put some basic details on her LinkedIn profile which she primarily uses for professional social networking, see figure 15.8.

The image shows a LinkedIn profile card for Polina Morphism. At the top left is a circular profile picture of a woman with dark hair and yellow flowers in it. To the right of the picture is a thumbnail image of a residential street with houses and trees. Below the profile picture is the name "Polina Morphism". Underneath the name is a description: "Student of Computing at The University of Poppletton with an interest in Object-Oriented Programming (OOP)". At the bottom of the card is the location "Upper Poppletton, England, United Kingdom · 1 connection". To the right of the profile picture is a purple square icon with a white minus sign, followed by the text "The University of Poppletton".

Figure 15.8: A pdf of Polina Morphism’s LinkedIn profile can be found at cdyf.me/Polly_Morphism.pdf, the original can be found at linkedin.com/in/polymorphism (you may be asked to login to see her profile)

15.3.8 Neil Pointer

Neil is a mature student who loves the C programming language, see figure 15.9. The Pointer family are sometimes misunderstood, but Neil compensates for this with his excellent memory management skills and efficiency. As well as twin sister Noelle, he also has three famous sisters, (Pointer et al., 1984) and a younger half-brother, Neil Pointer-Exception, from his fathers second marriage. Neil Pointer-Exception prefers Java. Noelle and Neil Pointer have seen many programming languages come and go, but their favourite will always be C. Like C, Neil is a child of the 1970s, easily “old enough to be your father” or your mother. Respect your elders! (Ritchie, 1993)

15.3.9 Bryn Hanby-Roberts

The last CV is a real one. Bryn kindly gave his permission to share it with you, see figure 15.10. Bryn graduated in 2016, his CV is longer as he has five years of



Figure 15.9: Neil Pointer’s full CV can be viewed at cdyf.me/Neil_Pointer.pdf

experience under his belt but it provides a useful counterpoint to the examples above. Thanks Bryn.



Figure 15.10: Bryn Hanby-Roberts full CV can be viewed at cdyf.me/brynpdf a snapshot taken in 2021 from brynp.co.uk

15.4 Sample CoolTech Job advert

We’re looking for bright and geeky graduates to join our software engineering team. No experience is required, and many of our successful applicants have never programmed before. If you think logically and enjoy problem solving, then you have the potential to become a great developer.

A career at CoolTech will challenge you every day. In your first few weeks you will be solving real-world problems as you help to develop software used by professionals across the world.

You’ll be part of an agile development team, working on one of the largest real-time databases in the world. You’ll work on a wide variety of projects, ranging from Artificial Intelligence assisting clinicians with early diagnosis of cancer to an iOS app helping patients manage their diabetes.

Developers at CoolTech are involved in the full software cycle, and work closely with all teams across the company to scope out new projects as they design, develop and deploy our products.

Chapter 16

Moving your future

Finding a job often involves moving to a different city or even a different country. So a simple way to improve your job prospects is to broaden your job search (chapter 8) and consider all the alternatives, geographically and otherwise. Some of the alternative options are outlined in chapter 9. This chapter looks at some UK alternatives to London.



Figure 16.1: Like many capital cities, London dominates the economy of its country. There are clearly some fantastic employers offering great opportunities in London but they do not represent everything that is on offer in the UK. There are plenty of good career options outside of capital cities like London if you don't want to live and work in a huge metropolis. The transition to remote and hybrid working also means you don't **have** to live in a large city. Not just London Image by Sharon Dale (Dale, 2018)

16.1 Hit the South?

For example, if you're seeking work or further study in the United Kingdom, there are lots of possibilities in Loxbridge (**London**, **Oxford** and **Cambridge**).

While Loxbridge undoubtedly offers many fantastic opportunities for your professional growth and development, it does not represent *all* of the best opportunities that exist in the UK see figure 16.1.

16.2 Hit the North!

If you're looking for a job in the UK you don't have to exclusively target the South (London) to find top employers, there are plenty located in the North too: Hit the North. (Smith et al., 1987)

Looking for Northern jobs in the UK?

See a partial list of tech employers at git.io/manc

Northerners have often argued that Northern England offers a better quality of life than London we couldn't possibly comment other than to say its horses for courses. The North West Tech Community calendar, provides a window on (and networking opportunities with) many of employers based in the North West technw.uk if you want to find out more. As Sharon Dale once said: its NotJustLondon (Dale, 2018)

So, it's wrong to assume that capital cities like London are where *all* the best opportunities are. That's true in the UK and also in many other countries too, and the rise of working from home during the pandemic has changed our relationship with cities. You might need to consider moving your future.

This is a partial list of employers in the North West of England (aka the Northern Powerhouse) that recruit Computer Science students. This is not a comprehensive list of all tech companies in the North West, but will give you a quick flavour of employers in the Manchester, Leeds, Liverpool, Sheffield and Northern England.

16.3 Hit the East and West

It's not just the North or the South either: You should also hit the East and the West too, don't just look at big cities. Don't just look at big employers. Look beyond the obvious, chapter 9 may help broaden your job search horizons.

In addition to these techuk.org and technation.io provide more information on technology businesses outside of London.

16.4 Summarising moving your future

The best jobs are not always in the obvious places. The myth that all the best graduate jobs are in London (or Paris, Berlin, Silly Valley etc) is clearly not the case, see figure 16.2.

Myth 4: 'All the best graduate jobs are in London'



Figure 16.2: It's a myth that all the best graduate jobs are in London and other capital cities or centres for technical innovation like Silicon Valley, Silicon Alley or Silicon Fen. For some more misleading myths see the universitiesuk.ac.uk report on *Busting Graduate Job Myths* introduced in section 1.6. (Ball, 2022a)

So a simple way to broaden your future and open up more possibilities is to consider all the geographical options. Move your future by looking beyond the obvious:

- Hit the North!
- Hit the South!
- Hit the East!
- Hit the West!

Chapter 17

Enjoying your future

Music is deeply connected to your mental health and your sense of wellbeing that we discussed in chapter 3. Music has the ability to comfort and connect you to others and yourself by providing motivation, gratification and empathy. There's plenty of scientific evidence that listening to music can enhance your mood. (Zatorre et al., 2007; McGilchrist, 2011)



Figure 17.1: Enjoying your future sketch by Visual Thinkery is licensed under CC-BY-ND

Besides entertaining you and lifting your spirits, there's also lot of wisdom captured in the lyrics of many songs too, offering sound advice on how you can live your life. If that wasn't enough to convince you that music is important for

your future, there's no shortage of links between mathematics and music too. (Harkleroad, 2006; Fauvel et al., 2006)

This chapter explores the musical references scattered throughout in this guide-book in a series of playlists which are accompanied with some explanation and context to help you enjoy them.

17.1 The Coder's playlist

In *Hearing your Future* (chapter 20) we ask our guests (coders) to recommend a tune that is important to them. These are gathered together in *The Coder's Playlist* and incremented by one with every new episode of the podcast. In the words of Arthur O'Shaughnessy, figure 17.2: We are the music makers and we are the dreamers of dreams. Wandering by lone sea-breakers, and sitting by desolate streams; World-losers and world-forsakers, On whom the pale moon gleams: Yet we are the movers and shakers of the world for ever, it seems.

We are the music makers,
And we are the dreamers
of dreams,
Wandering by lone sea-breakers,
And sitting by desolate streams;
—
World-losers and world-forsakers,
On whom the pale moon gleams:
Yet we are the movers and shakers
Of the world for ever, it seems

Ode (poem)

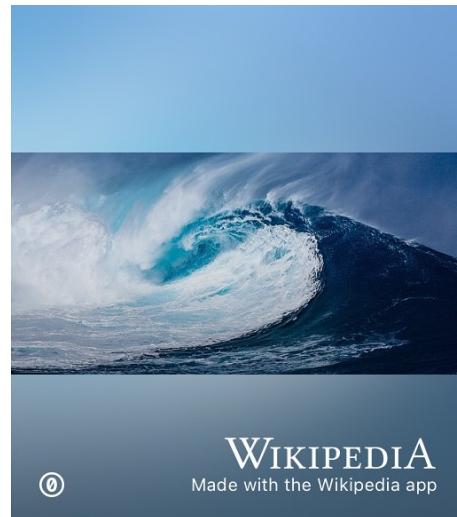


Figure 17.2: “We are the music makers and we are the dreamers of dreams” is a much quoted and sampled lyric from *Ode*, a poem written in the 19th Century (O’Shaughnessy, 1874). The poem is often credited with being the origin of the now widely used phrase “movers and shakers”. *Ode* was first re-used in music by Edward Elgar in *The Music Makers* (Elgar, 1912), and subsequently quoted by the character Willy Wonka in a film adaptation (Stuart, 1971) of the children’s novel *Charlie and the Chocolate Factory*. (Dahl, 1964) The Willy Wonka sample voiced by Gene Wilder has subsequently appeared in many recordings by artists such as 808 State (Massey et al., 1991) and Aphex Twin (James, 1992). Public domain picture of an ocean wave by unknown author on Wikimedia Commons w.wiki/5QW5 adapted using the Wikipedia App

- **Do say:** It's Music++
- **Don't say:** Music isn't important to me. You don't really care for music do ya? (Cohen, 1984)
- **Listen** at youtube.com/playlist?list=PLyxEvjuBJhvDi4guctovrW3ncSgqoVik_
- **Browse** the playlist notes in section 17.1.1

17.1.1 Playlist notes

Some notes to help you enjoy music on the coder's playlist:

1. "Anything" (we chose *No Idea*) by rapper Don Toliver was chosen by Jason, see chapter 23. (Toliver et al., 2019)
2. *LOUD* by singer Sofia Carson was chosen by Carmen, see chapter 24. (Carson, 2022)
3. *November Rain* by rockers Guns N' Roses was chosen by Sneha, see chapter 25. (Rose, 1992)
4. *Uptown Funk* by Mark Ronson is in this playlist because Aidan Blowers use of this music in his teaching has inspired me, see section 0.9.24 Thanks Aidan (Ronson and Mars, 2014; Blowers, 2015)
5. *Iguana* by singer Inna was chosen by Alice, see chapter 26. (Apostoleanu, 2018)
6. *As It Was* by Harry Styles was chosen by Alice, who managed to sneak two songs onto the playlist, see chapter 26. (Styles et al., 2022)

17.2 Turing's electro-playlist

This playlist celebrates electronica and other computationally influenced music. In 1951, engineers of the Ferranti Mark 1 computer, implemented a `hoot()` function that was used to sound a prompt when input or action was required. (Turing, 1950) Alan Turing realised that this could be reprogrammed to make different notes and created a routine that played a note of a specified pitch. (Copeland and Long, 2017)

Using Turing's code the computer scientist Christopher Strachey wrote programs to play music. Strachey's first success was a version of the UK's national anthem *God Save the King* (Anon, 1745) which was played at the end of what is very probably the first video ever game created, draughts shown in figure 17.3. (Link, 2012) The earliest existing recording of computer-generated music was made in 1951 by the BBC shown in figure ??.

\begin{figure}

In the recording you can hear the Ferranti Mark 1 computer playing three songs:

1. *God Save the King* (Anon, 1745)
2. *Baa, Baa, Black Sheep* (Anon, 1744)
3. *In the Mood* (Manone, 1935)

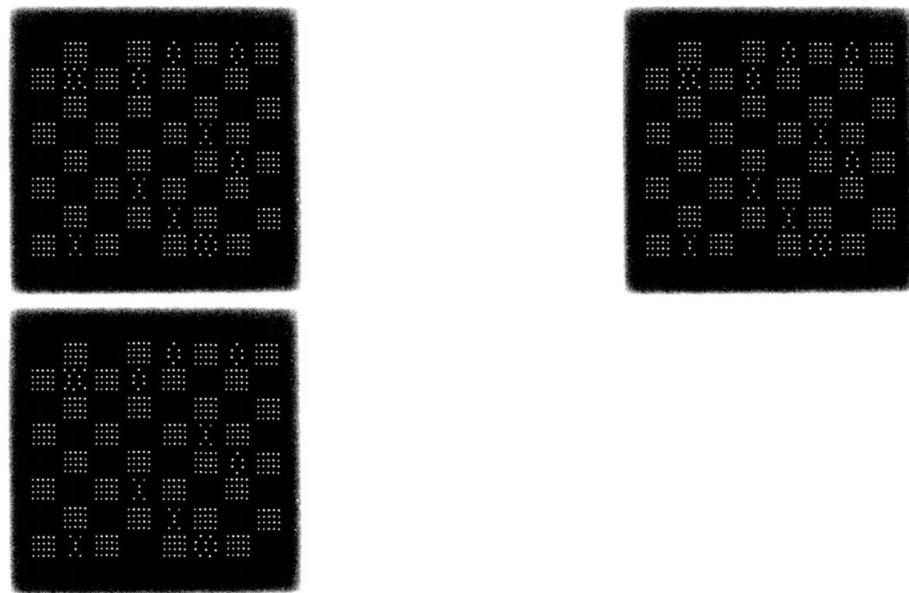


Figure 17.3: Is this the worlds first video game? Christopher Strachey's draughts game displayed on a cathode ray tube in 1952. When the game finished, the Ferranti Mark 1 computer which ran the game, played a tune. Public domain image from Wikimedia Commons at [w.wiki/5UJs](https://commons.wikimedia.org/wiki/File:draughts_1952.jpg)

You can listen this music which was restored by Jack Copeland and Jason Long in 2016. (Copeland and Long, 2016)

Computers have been changing the way we create, distribute and listen to music ever since. Sing like no one is listening, love like you never been hurt and dance like no one is watching:

- **Do say:** Turn it up DJ!
- **Don't say:** Why can't they play *real* musical instruments, you know, analogue ones?
- **Listen** at youtube.com/playlist?list=PLyxEvjuBJhvCC2vbFMosi5lIXPPSKwck

17.3 The Lancashire playlist

This playlist starts and ends with one of the best known bands on the planet: The Beatles shown in figure 17.4. You will find plenty of other Lancashire lads, lasses and even Lucifer in here too. (Smith and Hanley, 1986)

Which cities are actually *in* Lancashire, I hear you say? Don't let bureaucrats trick you into believing that:

- Manchester is in a place called “*Greater Manchester*”
- Liverpool is in a place called “*Merseyside*”

These are *meaningless* bureaucratic abstractions, not **real places** that people identify with! Liverpool and Manchester have always been, and will always be, in Lancashire. Aye! (Groom, 2022)

Bursting with Lancastrian Red Roses (see figure 17.6) this playlist will educate and entertain you in equal measure, while my keyboard gently weeps (Harrison, 1968). To paraphrase Robbie Williams shown in figure 17.5:

- **Do say:** Chippy tea, chippy tea, I wants a chippy tea! (Thresher et al., 2007; Carroll, 2018)
- **Don't say:** But Liverpool and Manchester aren't even *in* Lancashire are they? (Hull, 2019)
- **Do say:** This land's the place I love and here I'll stay (Marsden, 1964)
- **Listen** at youtube.com/playlist?list=PLyxEvjuBJhvDvG0TRu63n3i12kR89niFJ

17.4 The Dad-rock playlist

Using the medium of rock, this playlist grumpily reminds you that “*when I was your age*” music was much better and it's all gone downhill since the good old days. Nostalgia isn't what it used to be but these classics were made by *true* artists using *proper* musical instruments, see figure 17.7. None of this social media powered digital nonsense thank you very much! (Lanier, 2018)



Figure 17.4: John, Paul, Ringo and George (aka The Beatles) are from Liverpool in Lancashire. Public domain image of The Beatles in New York City in 1964 via Wikimedia Commons at w.wiki/5UJy

Williams and Chambers were inspired to write a 'Who-esque' song after watching the Rolling Stones film Rock and Roll Circus together

Let Me Entertain You
(Robbie Williams song)



Figure 17.5: Robbie Williams and Guy Chambers were inspired to write *Let Me Entertain You* after watching *The Rolling Stones Rock and Roll Circus*. (Lindsay-Hogg, 1996) “Come on, let me entertain you! Come on, let me educate you!” (Williams and Chambers, 1997)

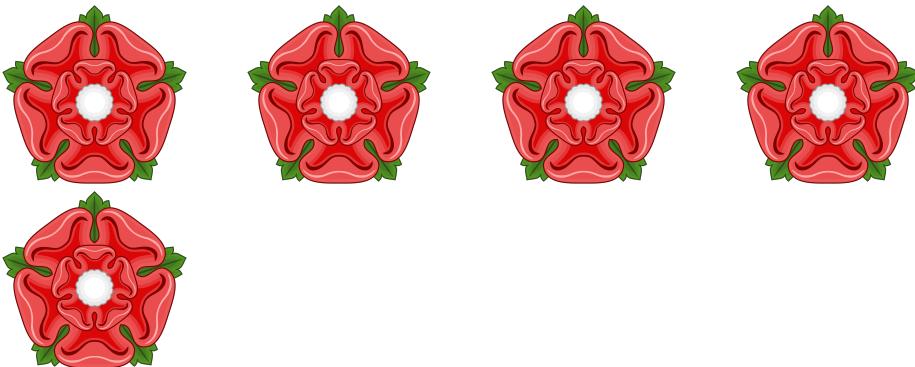


Figure 17.6: The Red Rose of Lancaster is a symbol of the county of Lancashire, first adopted by the House of Lancaster in the 14th century. There's no 14th century music in the Lancashire playlist, but there's plenty of red roses. CC BY-SA image of a red rose by Sodacan on Wikimedia Commons w.wiki/5Y4Z

he is widely regarded as one of the most influential electric guitarists in the history of popular music, and one of the most celebrated musicians of the 20th century

Jimi Hendrix



Figure 17.7: Are you experienced in Dad-rock? (Hendrix, 1967) Definitions vary, but there's plenty of men who believe this playlist includes some of the best music ever made. By real musicians. With real musical instruments. Public domain image of Jimi Hendrix performing on the *Hoepla* show in 1967 by A. Vente via Wikimedia Commons w.wiki/5WHj adapted using the Wikipedia App

Cherished by men of a certain age, the Dad-rock playlist thrives on past glories but is a shadow of its former self. Proponents of Dad-rock can't understand why more people don't enjoy these seminal tunes of yesteryear (Mitchum, 2019; Rogers, 2008):

- **Do say:** Expiry date: What expiry date? Classic music like this is timeless!
- **Don't say:** Oh no, not this again!
- **Listen** at youtube.com/playlist?list=PLyxEvjuBJhvDUyr9hPvUP3IeriN8aaTW0
- **Note:** we've broadened the definition of Dad-rock to include *any* music that your Dad might have listened to "*when he was your age*" ...

17.5 The Indie rock playlist

Wetter and muddier than the traditional summer downpour at the Glastonbury Festival of Contemporary Performing Arts in Somerset (see figure 17.8), this playlist encourages you to swing your cardigan mournfully while shoegazing and wondering if its *ever* going to stop raining, either metaphorically or meteorologically:

a five-day festival of
contemporary
performing arts that
takes place in Pilton,
Somerset, England

Glastonbury Festival



Figure 17.8: "Why does it always rain on me? Is it because I lied when I was seventeen *I went to Glastonbury festival?*" (Healy, 1999) If you're only happy when it rains, if you're only happy when it's complicated, then the indie rock playlist is for you. (Manson et al., 1995) All art is borne of suffering and this playlist helps you wallow in it. Pour your misery down on me! Picture taken from the highest point at Glastonbury festival in 2016 by Czampal on Wikimedia Commons w.wiki/5WKn adapted using the Wikipedia App

- **Do say:** It's indie rock and roll for me! (Flowers et al., 2004)

- **Don't say:** Cheer up hun, life's not *that* bad is it? (Fonarow, 2010)
- **Listen** at youtube.com/playlist?list=PLyxEvjuBJhvBUO8jaKvJtZLuXfnKc8iz3

17.6 Thanks Jim

Thanks to Jim Miles for the descriptions of the birth of electronic music installed in the Kilburn building Atlas suite shown in figure 17.9. Some of the text from these displays has been adapted and re-used in section 17.2. (Miles et al., 2022)



Figure 17.9: Panorama of the Atlas suite in the Kilburn building, Manchester which exhibits hardware from several Manchester computers such as the Atlas computer. Picture by Yours Truly.

Chapter 18

Actioning your future

Employers are often more interested in what you have *done*, rather than what you just *know*. Your actions are a key part of your story we discussed in section 6.7.2. A simple technique for emphasising the **action** in your stories is to lead descriptions of your projects, education and experience with carefully chosen **verbs**, see section 6.7.4 for examples.

18.1 Your actions define your impact

Your actions define your impact, see figure 18.1. What stories you can tell of your actions to date? What verbs best describe how you achieved a result or had an impact? What was the context, action, result and evidence (CARE) we discussed in section 6.7.2 of each (short) story?

By leading with verbs you will highlight what you have *actually done* and how you did it, rather than what you know. See the verbs first section 6.7.4 of chapter 6 *debugging your future*.

18.2 What you will learn

By the end of this chapter you will be able to:

1. Emphasise your *actions* when describing your education, projects and experience
2. Reflect on
 - what skills you already have
 - what skills you need to develop
3. Demonstrate those skills explicitly and quickly in job applications



Figure 18.1: What action have you taken and what stories can you tell about the results and your impact? What are the best verbs for highlighting your actions? Your actions define your impact by Visual Thinkery is licensed under CC-BY-ND via Angela Maiers

18.3 Breakpoints

Let's pause here. Insert a breakpoint in your code and slowly step through it so we can examine the current values of your variables and parameters.

* PAUSE

Quickly scan your CV, covering letter or application form for VERBS:

- Where are the verbs?
 - buried deep in long sections of prose? **OR**
 - prominently leading descriptions of your activities?
- Have you over-used certain verbs (like **worked** or **assisted** for example) or been repetitive (like over-using **developed** see alternatives in section 18.5)
- How can you increase the variety of verbs you have used (without exaggerating or lying)?
- Which verbs are stronger than others and why?
- Are there any categories of verbs you can't provide evidence for, such as leadership (see section 18.6) or influencing (see section 18.11)?
 - What activities or projects could you do that would help you develop these missing skills?

*** RESUME**

18.4 Team verbs

Some verbs to demonstrate how you have worked and communicated with others in a team.

- **administered**
- **advised**
- **assisted**
- **attended** (but show outcomes)
- **coached**
- **collaborated**
- **contributed**
- **encouraged**
- **explained**
- **instructed** (if you helped others)
- **interviewed**
- **organised**
- **participated**
- **presented**
- **recommended**
- **recruited**
- **shadowed** (e.g. work shadowing)
- **suggested**
- **volunteered**

18.5 Engineering verbs

Verbs to demonstrate your engineering and technical skills.

- **adapted** (e.g. new features)
- **added** (e.g. new features)
- **analysed** (e.g. the requirements)
- **architected**
- **assigned** (e.g. bugs to team members)
- **automated** (e.g. builds and tests etc)
- **built**
- **branched** (e.g. git)
- **configured**
- **designed** (e.g. greenfield software development)
- **cloned** (e.g. git)
- **debugged** (e.g. brownfield development)
- **developed**
- **deployed**

- documented
- experimented
- gathered (e.g. requirements)
- implemented (e.g. an algorithm)
- installed
- integrated (e.g. different systems)
- made
- merged (e.g. git)
- migrated
- modified
- refactored
- solved
- specified
- upgraded
- tested

18.6 Leadership verbs

Some verbs to demonstrate how you have used your initiative and taken the lead:

- argued (persuasive speaking or writing)
- established
- created
- decided (you've had the power to make or influence decisions)
- devised
- directed
- facilitated
- initiated
- introduced
- launched
- led
- managed
- mentored (if you've helped by sharing your skills and knowledge)
- motivated
- supervised
- transformed (you changed something for the better)

18.7 Improving verbs

Verbs that demonstrate how you have improved a situation by taking responsibility for something:

- delivered
- completed (if you finished something)

- edited
- enhanced
- generated
- increased (make sure you quantify it, see section 6.7.2)
- learned you are a student after all!
- refined
- resolved (a conflict)
- saved (money, time, resources etc)

18.8 Scientific verbs

Verbs that demonstrate your analytical and scientific skills

- assessed
- calculated
- discovered
- estimated
- evaluated
- identified
- interpreted
- investigated
- measured
- modelled (in a computational or mathematical sense)
- proved
- quantified (for example in benchmarking)
- researched

- reviewed
- studied
- tested

18.9 Winning verbs

Verbs for demonstrating your achievements and honours

- achieved
- attained
- awarded
- nominated
- recommended
- selected (you were chosen for something)
- mastered
- won

18.10 Organising verbs

Verbs to demonstrate your planning and organisational skills:

- arranged
- prepared
- scheduled
- organised
- planned
- produced (making things, not just software)
- revised

18.11 Influential verbs

Verbs that demonstrate how you have influenced and persuaded others:

- bought (if you've had purchasing power)
- campaigned
- guided
- demonstrated
- illustrated (if you have graphical skills for example)
- influenced (could even include social media influencing)
- liaised
- negotiated
- marketed
- mediated
- promoted
- presented
- publicised
- sold (an idea, product or service)
- authored (or co-authored)
- written

18.12 Summarising your actions

Too long, didn't read (TL;DR)? Here's a summary:

Actions speak louder than words, or as suffragette and political activist Emeline Pankhurst frequently said “Deeds not Words”, see figure 18.2. Your CV needs to emphasise your deeds and actions using words. Those words are **verbs**.

Leading with **verbs** is a simple but powerful technique that enables you to provide evidence (rather than assertion) for the skills and knowledge you have. Choose your **verbs** carefully. Which verbs are missing from your CV? These verbs can help you identify gaps in your professional and personal development.

“deeds, not words”

Emmeline Pankhurst

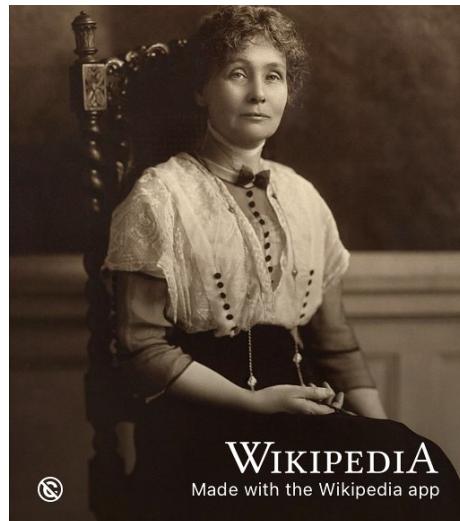


Figure 18.2: “Deeds not words” was the rallying cry of suffragette Emmeline Pankhurst. Emphasise the deeds (actions) on your CV by leading your stories with carefully chosen **verbs**. Public domain image of Emmeline Pankhurst by Richard Gordon Matzene restored by Adam Cuerden on Wikimedia Commons [w.wiki/3bPa](https://commons.wikimedia.org/w/index.php?title=File:Emmeline_Pankhurst_by_Richard_Gordon_Matzene.jpg&oldid=3131334) and adapted using the Wikipedia app

Chapter 19

Scheduling your future

You might find it a bit scary thinking about your future. You might be tempted to procrastinate making important decisions about your future, see figure 19.1. There is a risk of thrashing or getting stuck in a **do-nothing** or **busy waiting** loop. This guidebook is here to help you break out of that loop. One way to breakout of an unproductive loop is to schedule some time every week where you work on personal development and job applications. Doing good applications takes time and you'll probably find you can't do as many applications as you might like.

If you're a University of Manchester student, the live *Coding your Future* (COMP2CARS) workshops sessions are also here to help, every Monday at 5pm. COMP2CARS complements the second year tutorials (COMP2TUT) at the University of Manchester and takes place in the same slot as COMP2TUT when you meet your personal tutor. See your timetable at timetables.manchester.ac.uk.

19.1 Week 11: Meeting

Monday 6th December at 5pm:

- *Either* COMP2TUT: Meet your tutor see studentnet.cs.manchester.ac.uk/ugt/tutorgroups.php?level=2
- *Or* COMP2CARS: self study

19.2 Week 12: Meeting

Monday 13th December at 5pm:

- *Either* COMP2CARS: self study
- *or* COMP2TUT: Meet your tutor (if you didn't meet them last week) see studentnet.cs.manchester.ac.uk/ugt/tutorgroups.php?level=2

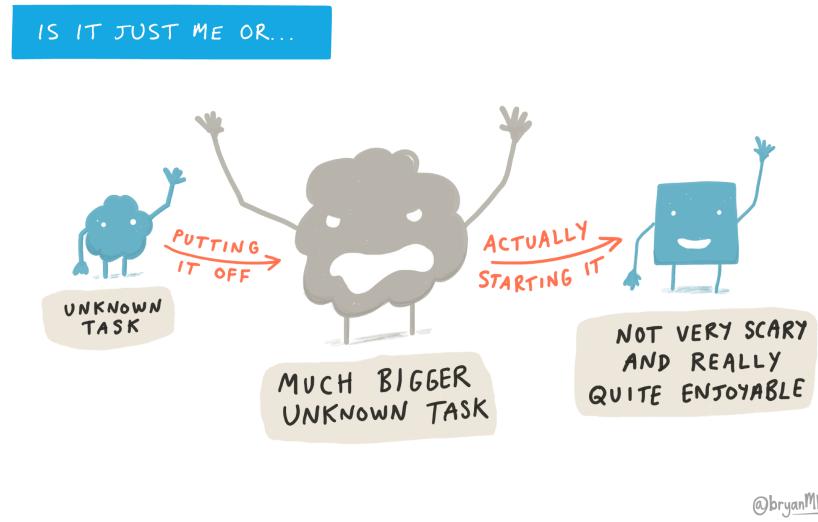


Figure 19.1: The biggest waste of time is the time spent not getting started on a project. Your future might seem big and unknown but it's really not as scary as you might think and getting started can be surprisingly enjoyable. New Project? Every time... by Visual Thinkery is licenced under CC-BY-ND

19.3 Discussing your Future

The accompanying online forum for course discussion will be at pizza.com/class/l8t6l0le6vu2lg

Part IV

CODING

Chapter 20

Hearing your future

Hearing your Future is the Coding your Future podcast which accompanies this guidebook. We talk to graduands and graduates of computer science to hear more about their journey from student to professional. What happens after graduation like the one shown in figure 20.1? What comes next?

Listen in to the episodes below, or read the transcripts, to find out more about how these students are coding their future:

- What's their story?
- How did they get to where they are?
- What obstacles have they faced and how did they overcome them?
- Where are they planning to go next?
- What advice would they offer to their former selves and fellow students?

Find out more by listening here or subscribing wherever you get your podcasts...

20.1 Subscribing to your future

You can subscribe (for free!) by following one of the links below to your favourite podcast tool. Alternatively you can search for **Coding your Future** or **Hearing your Future** in your podcast application. You will see the logo shown in figure 20.2 when you find it.

- Spotify: spoti.fi/3SFqaYk
- Apple: apple.co/3JFV5Qe
- Amazon Music / Audible: amzn.to/3OOaS0G
- Google podcasts: podcasts.google.com/feed/aHR...
- Deezer: deezer.com/en/show/3839607
- Podcast homepage: codingyourfuture.libsyn.com



Figure 20.1: Silly hat? Silly frock? Wearing your best outfit? It must be time for graduation! What comes next? *Hearing your future* interviews current and former students to find out more about how they got to where they are and where they are going to next. Picture of obligatory hat throwing outside the Samuel Alexander building in 2018. Photobombing by Gavin Brown

If all else fails, you can cut-and-paste the raw `rss` feed URL below into the relevant settings on your podcast app:

<http://feeds.libsyn.com/299189/rss>

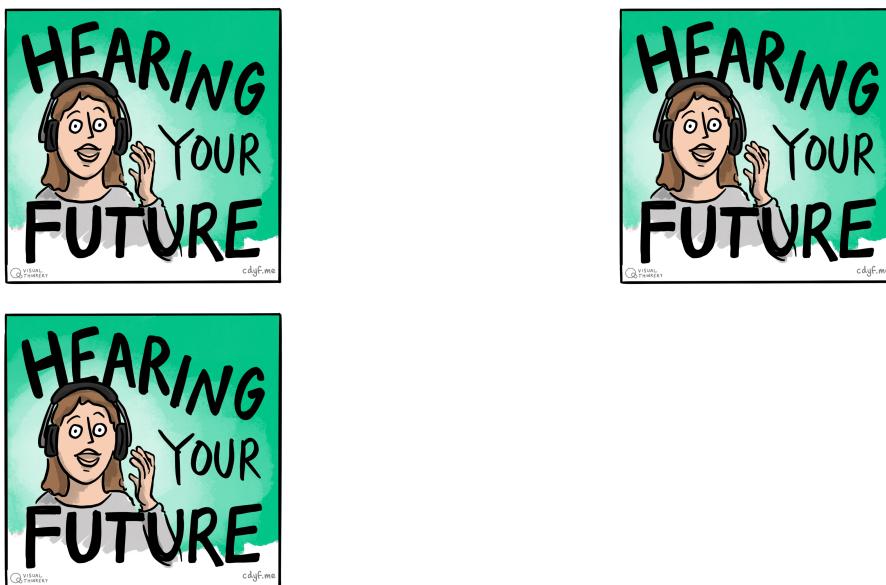


Figure 20.2: *Hearing your future* is the Coding your Future podcast. Subscribe wherever you get your podcasts. Hearing sketch by Visual Thinkery is licensed under CC-BY-ND

If you'd like to be a guest on the show, see section 20.7

20.2 Episode 5: Sneha Kandane

Interview with Sneha Kandane, see figure 20.3 and the transcript and show notes in chapter 25.

Listen to the episode by clicking Play below, or subscribe wherever you get your podcasts, see section 20.1.

20.3 Episode 4: Carmen Faura Práxedes

Interview with Carmen Faura Práxedes, see figure 20.4 and the transcript and show notes in chapter 24

Listen to the episode by clicking Play below, or subscribe wherever you get your podcasts, see section 20.1.

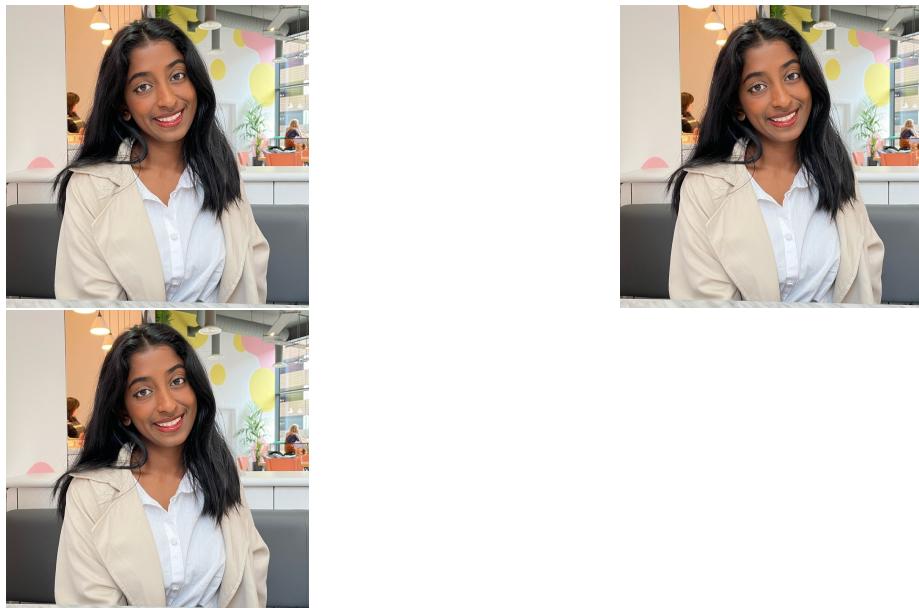


Figure 20.3: Sneha Kandane. Picture reused from [linkedin.com/in/sneha-kandane-931346183](https://www.linkedin.com/in/sneha-kandane-931346183) with permission, thanks Sneha.



Figure 20.4: Carmen Faura Práxedes. Picture reused from [linkedin.com/in/carmen-faura](https://www.linkedin.com/in/carmen-faura) with permission, thanks Carmen.

20.4 Episode 3: Brian Yim Tam

Interview with Brian Yim Tam, see figure 20.5 and the transcript and show notes in chapter 22



Figure 20.5: Brian Yim Tam. Picture reused with permission from [linkedin.com/in/byt411](https://www.linkedin.com/in/byt411), thanks Brian.

Listen to the episode by clicking Play below, or subscribe wherever you get your podcasts, see section 20.1.

20.5 Episode 2: Jason Ozuzu

Interview with Jason Ozuzu, see figure 20.6 and the transcript in chapter 23.

Listen to the episode by clicking Play below, or subscribe wherever you get your podcasts, see section 20.1.

20.6 Episode 1: Raluca Cruceru

Interview with Raluca Cruceru, see figure 20.7 and the transcript and show notes in chapter 21

Listen to the episode by clicking Play below, or subscribe wherever you get your podcasts, see section 20.1.

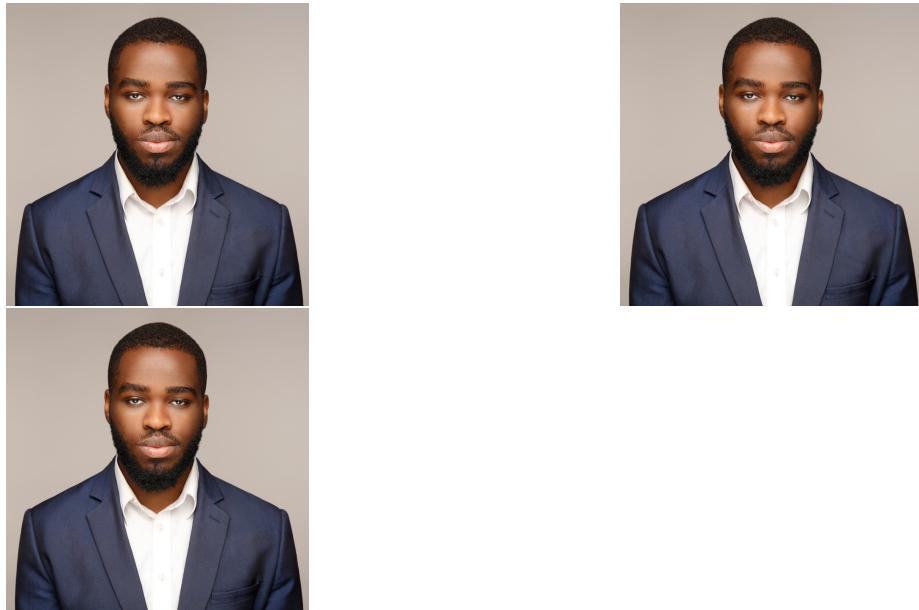


Figure 20.6: Jason Ozuzu. Picture reused with permission from linkedin.com/in/jason-ozuzu-a87049173, thanks Jason.

20.7 Episode *x*: it could be YOU!

YOUR FUTURE WANTS YOU: Join your future! If you'd like to be interviewed for the show, get in touch, see figure 20.8. Besides interviewing current graduates, I'm interested in speaking to former graduates, especially if you:

- completed a placement or internship(s) as an undergraduate or postgraduate
- consider yourself to be part of a minority or under-represented group

I'm also interested in speaking to people who graduated a while back, not just this years graduates, but any back to 1968. (Brackenbury, 2005)

If you're preparing to come on the show, please have a think about the following questions:

20.7.1 What's your story, coding glory?

With apologies to the Noel Gallagher and Liam Gallagher, see figure 20.9:

“All your dreams are made, when you’re chained to the tracker and the software trade... What’s your story, coding glory?”

What's your origin story?

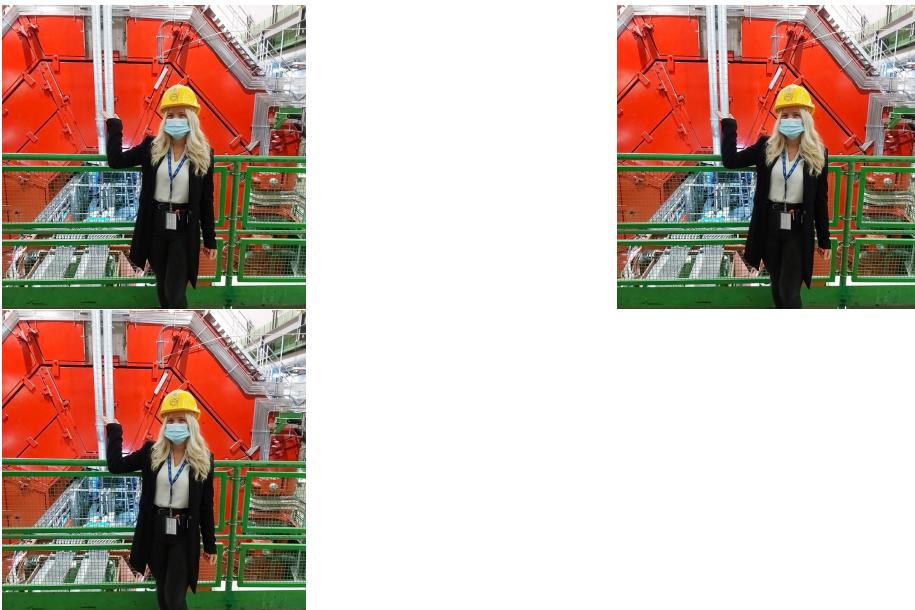


Figure 20.7: Raluca Cruceru is a software engineer at CERN, standing here in front of the experiment she works on: A Large Ion Collider Experiment (ALICE) part of the Large Hadron Collider at CERN, see careers.cern/Raluca. Picture reused with permission from linkedin.com/in/raluca-cruceru, thanks Raluca.



Figure 20.8: Your country future wants YOU. Join your future. If you are a former student of Computer Science who'd like to appear on the show, get in touch. I'm especially interested to hear from students who did internships or placements before they graduated in Computer Science. Picture adapted from an original public domain image of the Lord Kitchener Wants You poster by Alfred Leete on Wikimedia Commons at w.wiki/3xvX

- What's your name and where do you come from? (Black, 1985)
- Why did you choose to study computer science?
- Which organisation were you employed by, why and how did you choose them
- What were the main obstacles you faced finding employment and how did you overcome them?
- Tell us about your roles and responsibilities within the organisation
- How did you find the job and what other jobs did they look for?
- What were the main things you learned on placement?
- What was the most enjoyable or rewarding part of working for your employer?
- What did you do for your final year project?
- If you graduated previously, what has been your career path to date?
- What comes next?

**(What's the Story)
Morning Glory? was
inspired by Noel's friend
Melissa answering the
phone with said phrase**

**(What's the Story)
Morning Glory?**



Figure 20.9: (What's your Story) Morning Coding Glory? (Gallagher, 1995). CC BY portrait of Noel Gallagher by alterna2.com on Wikimedia Commons w.wiki/3bimy adapted using the Wikipedia app

20.7.2 Minority report

This is an optional question taken from the title of the Philip K. Dick novel *The Minority Report* and subsequent film adaptation, see figure 20.10. Minority report asks our guests:

- Do you consider yourself to be a member of an under-represented, minority or otherwise marginalised group?
- If so which one(s) are you happy to discuss, see section 2.2.1?

- What has your experience been of being in a minority at University and in the workplace?
- How can the teaching and learning of computer science be made more equitable, diverse and inclusive for members of your minority group(s)?
- What else can universities do to make campuses more welcoming to members of your minority group(s)?
- What more can employers do to make workplaces more welcoming to members of your minority group(s)?

"The Minority Report" is a 1956 science fiction novella by American writer Philip K. Dick, first published in *Fantastic Universe*.

The Minority Report

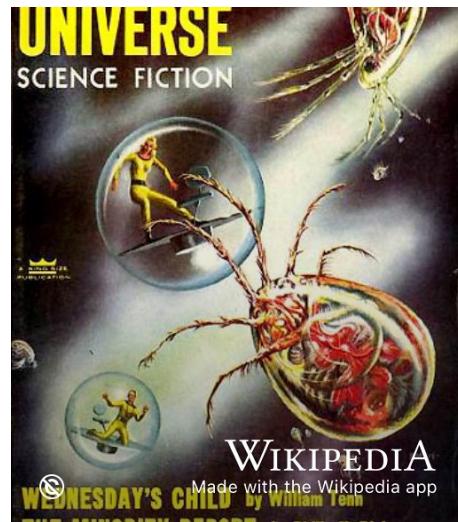


Figure 20.10: Are you in a minority? Inspired by the book (Dick, 1956) and its film adaptation (Spielberg, 2002), we'll hear from some of our minorities. *The Minority Report* is short story by Philip K. Dick first published by *Fantastic Universe* magazine in 1956. What is it like being in minority and how could universities and employers be more welcoming to your minority group(s)? Public domain image of magazine cover by Ed Emshwiller via Wikimedia Commons w.wiki/5Swn adapted using the Wikipedia app

20.7.3 One tune, one podcast and one film

We love our music, films and podcasts and make gratuitous references to them throughout *Coding your Future*. Critics say our playlist in chapter 17 is clogged up with:

- Lancashire boy bands, see section 17.3
- DadRock and BritPop, see section 17.4
- cardigan-swinging, shoe-gazing, glastonbury-going indie music, see section 17.5
- music made with digital computers rather than analogue instruments, see section 17.2

- other antiquities from a bygone era (Rogers, 2008)

So, can you recommend a:

- **TUNE!** One tune to rejuvenate or internationalise up our playlists. Why is it important to you?
 - Perhaps it makes you relax, feel happy or comforted?
 - Maybe it reminds you of special people, a special time or special place in your life?
 - We'll add it to *The Coder's Playlist* in section 17.1
- **PODCAST:** One podcast (or audiobook) you've enjoyed that you'd recommend people listen or subscribe to
- **FILM:** One film we've got to go and watch right now, old or new

20.7.4 Time traveller

Do you wish that you knew what you know now, when you were younger? Just like Ronnie Lane, Ronnie Wood and Rod Stewart, we all probably feel that way sometimes, see figure 20.11. So, if you could travel back in time to meet the younger you, just starting University:

- What would you tell your former self, now that you are older and wiser?
- What would you tell current students to help them get the most of their preciously short time at University?

**"I wish that I knew what I
know now, when I was
younger!"**

Ooh La La (Faces song)



Figure 20.11: Poor old grandad I laughed at all his words but I wish that I knew what I know now, when I was younger, see Ooh La La. (Lane and Wood, 1973) Do you wish that you knew what you know now, when you were younger? What would you tell your younger self about getting the most from University life?

20.8 Everything I had to know, I heard it on my radio

We'll leave the last words on podcasting to *The Buggles* and *Queen*. Podcasting is an evolution and mutation of radio but did video kill the radio star? (Downes et al., 1979) What if everything you had to know, you could hear it on your radio, see figure 20.12? Tune in to find out.

A nostalgic defence of
radio, it was a
commentary on
television overtaking
radio's popularity

Radio Ga Ga

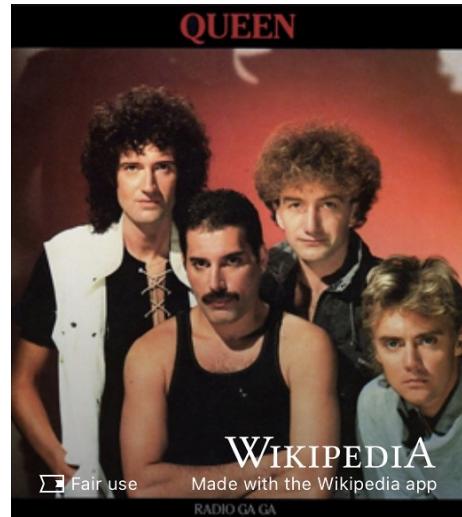


Figure 20.12: All we hear is Radio Ga Ga but everything I had to know, I heard it on my radio. (Taylor, 1984) So don't become some background noise, a backdrop to the girls and boys, who just don't know and just don't care and just complain when you're not there.

20.8.1 Thanks Jez

This podcast is inspired and supported by Jez Lloyd, host, producer and mastermind of the excellent CS@Manchester podcast, see figure 20.13. Thanks Jez!

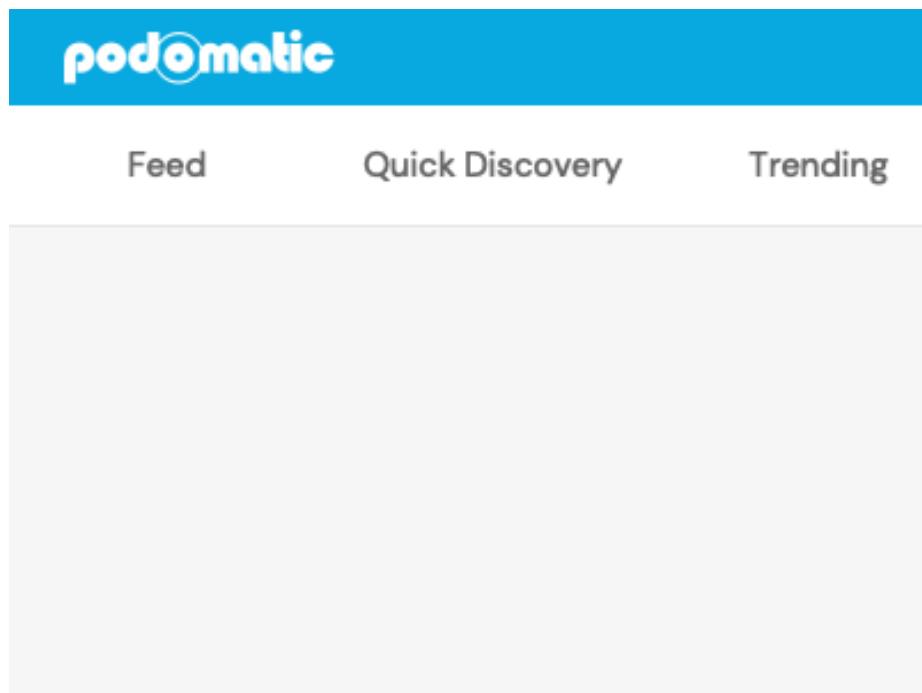


Figure 20.13: The CS@Manchester podcast was produced and hosted by Jez Lloyd and ran from 2015 to 2020. This podcast, *Hearing your future*, is following in all 38 of its audible footsteps, see podomatic.com/podcasts/cs-engagement

Chapter 21

Raluca

Meet Raluca Cruceru, shown in figure 21.1 she graduated with a BSc in Human-Computer Interaction with Industrial Experience in 2020.

Listen to the episode by clicking Play below, or subscribing wherever you get your podcasts, see section 20.1. An annotated and edited transcript of the audio is shown below.

21.1 What's your story Raluca?

My guest today is Raluca Cruceru. I want to find out a little bit more about your story and your journey from arriving here. Four years ago, is that right? Three and a half years ago? Yeah, it was it three and a half years now, isn't it? Can you tell us a little bit about yourself your name and where you're from?

Raluca: Yeah, of course. My name is Raluca Cruceru. I am 22 years old and originally from Romania. I came to the University of Manchester to study computer science. And here I am now in my final year,

about to graduate and go off into the big wide world. So I think it's always interested hearing what made what made people study computer science so what was it that made want to study it? While you were doing your exams, applying to Manchester and what why did you study why did you choose to study computer science as a subject? Not why Manchester necessarily, but why computer science?

Raluca: Well, I think that everything I've experienced before university connected together to my final decision, when I was a child, I was always interested into how computers work or what they work that they work, but of course at the time, everything was worried. But then in high school, it became more clear that I'm attracted



Figure 21.1: Raluca Cruceru is a software engineer at CERN, standing here in front of the experiment she works on: A Large Ion Collider Experiment (ALICE) part of the Large Hadron Collider at CERN, see careers.cern/Raluca. Picture reused with permission from linkedin.com/in/raluca-cruceru, thanks Raluca.

to this idea of technology and computers. My class was focused on mathematics and informatics. So therefore, I was always into this kind of information. The solving problems became natural for me I could spend hours doing that. So at the end of the high school, it was obvious that I wanted to do something concerning Computer Science and Mathematics. And it was a decision between mathematics and computer science. And I chose computer science because it felt like something that could be used in more circumstances in more industries, and more linked to the general technology. Yes.

So did you I guess, did you do you must have done a fair amount of computer science at high school did you do much in the way of coding in high school or was it more more focused on the mathematics?

Raluca: It was focused on the mathematics but then we also started to code in high school, and I think this is actually what helped me a lot to realise that they want you to do computer science. Also one of my exams that I chose at the at the end of my high school was boring informatics. I kind of fell in love with this idea. So yeah, yeah.

So, how did you go about deciding whether that was something that you wanted to do?

Raluca: Initially, I didn't know about this option to have one year experience in industry. But when I found about it, I was really excited because I think for myself, the hands on experience, very important. And of course, we did all the way behind and all the lessons but I really wanted to understand what software engineering job actually requires. So as soon as I found out about this option, I was motivated to take it. I started to search for possibilities.

21.2 Web Applications UK

So this typically, most people are looking for placement style that are getting their second year. So after you've done the first year, so what sort of what sort of places did you apply to and what you were looking at lots of different things. You weren't focused in on one, one or two places?

Raluca Well, to be honest, I was very focused on certain because I knew I wanted to work there. However, my I was very doubtful. Regarding my chances. I thought it was my only option. I applied for other companies in the UK. In places like CERN will expect you to have, or your application will stand out a lot more if you've got some experience like Yeah.

So you've done a summer job Web applications UK, which is now called, they've changed their name now to Koder.ly with a K. So, you've got a broad exposure

to different sorts of software engineering ideas, working in a team and that kind of stuff.

Raluca: Yeah, it really meant a lot because it's one thing to work on a lab at university to be in another thing to work on something that you know, can impact a whole number of people. So it makes you want to do a good job. You know, that you're, the stuff that you're writing is going to be used by hundreds, sometimes 1000s of people

I think that's one of the satisfying things about software engineering is knowing that what you've made can get ended up getting used by lots of people

Raluca: but stressful, stressful and exciting in my opinion, right? Yeah, there's, there's lots of places it can go wrong. Isn't that I think when you're building software,

21.3 CERN

okay, good. So, I think you mentioned you were particularly focused on CERN. I think they have two application rounds. Is that right? So you applied in the autumn?

Raluca: No, actually applied in the second one, from May onwards.

All right. So you didn't you didn't apply in the first round in autumn?

Raluca: In the second round in May,

I guess lots of people have heard a lot about CERN but what can tell us more about CERN as an organisation and what your what your role was, while you were a technical student there for a year?

Raluca: so the European organisation for Nuclear Research in Switzerland, and their main focus is on high energy physics research. They operate the largest particle physics laboratory in the world. The main focus is to unravel the mysteries of the universe, more specifically, how it was formed, and from there to understand how it will evolve in the future. And the thing that attracted me the most was the fact that they push the frontiers of not only science, but technology computing, and also medical technology. And maybe a lot of people know that the World Wide Web was created there in 1989 (see (Cailliau, 2014) for web history and also birth of touchscreen technology (Stumpe, 2014)) And it's interesting to know that I was to a university where we have a software that has no computers, and then go through the transition that created the World Wide Web. Like everything is connected. And, of course, I knew about since I was a child that I didn't really imagine that there would actually work there. It was just

like something to dream about when you go to sleep. But yeah, in the end, I got accepted. And I worked as a software engineer in the electrical engineering department. Basically, my task included building and maintaining software applications for the department for monitoring projects of cabling and fibre optics there. And I have several tasks to do, building technical panels that would extract and migrate information from different databases regarding this project. Then, a dashboard that was monitoring the project's evolution and generating statistics regarding the quantities of cables for the radiation dose intake. And then after that, those two pieces were published on the department website, which I also maintained. So as you can see, there were a lot of tests and I learned very much how to be able to handle multitasking that time.

The thing that thing that I think is perhaps most surprising about CERN is the fact that for every scientist they employ there are 10 engineers (see figure 21.2), and an awful lot of those engineers are software engineers, but you've also got the hardware and electrical engineering as well. So there's an awful lot of engineering that goes on.

CERN employs ten times
more engineers and
technicians than
research physicists

CERN

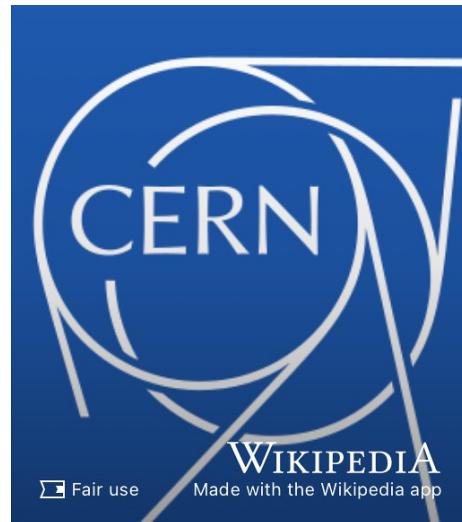


Figure 21.2: CERN employs ten times more engineers and technicians than physicists (Hull, 2020). Fair use image via Wikimedia Commons w.wiki/4qmF adapted using the Wikipedia app

Raluca: I mean, it's a big, it's a big machine underground, basically, isn't it at the end of the day. machines need lots of engineers. So there's lots of lots of stuff but I guess it was nice being part of a international community while you were there, because you're you're meeting people from all over literally all over the world and princi-

pally Europe, but you and all students and staff coming together in the community they created there? It's so beautiful. There are scientists and engineers from around the world. And the atmosphere is just great. You feel like everybody has the same purpose. And we have something to collaborate for, and something to work on together. It just brings people together from different domains is great.

So with that, so when you applied to CERN, what was your strategy because I know you have to write sort of a personal statement type thing where you describe your interests. So were there any things that you emphasise? Did you particularly tailor what you'd written to particular groups at CERN? Because there's lots of different groups certain doing software engineering is not just one, did you? Did you tailor it at all? Was it anything? Anything that you want, you know, how did you go about applying?

Raluca: The thing that I knew was that when you apply there's not a specific departments to apply to? So you're applying and then you were sent to a specific group. But yeah, I spent a lot of time in trying to make my application. Having having a personal touch. I was as honest as possible about why I wanted to work at CERN about what you personally I respect what they do. And of course, I try to talk enough and a lot about what makes me a good candidate for them. Yeah, I think we, in general, I think when applying people should put a lot of focus on details. Even if you think that something is not relevant to put the accent on. It can be actually something that makes you stand out. Even at the time, I was not sure how would be accepted even with all these efforts. I was still doubtful. I reminisce it takes quite a long time to do it as well. So every application and in particularly a certain application takes a long time to have to think quite hard about what you're being asked and what you're going to say. And yes, you never know if any of this is going to come to anything it can be quite demoralising sometimes filling in all these forms and writing all these personal statements when you never actually know if you'll get a reply. Yeah, because they asked you not only about what you can do, but more questions to see what kind of person you are. And what do you want to bring into the department we're working? Yeah, they have to pay a lot of attention.

So you were at CERN for, was it 12 months or were you there for slightly longer did you extend over the summer?

Raluca: Yes, it was initially one year but then I had the contract extension with three more months when they proposed that to me, I was more than happy to accept it. And of course, I had to confirm it with university. And luckily it was accepted.

Did you learn any French when you did you? Do you speak any French when you're out there and or was it mostly in English? That you were speaking?

Raluca: Mostly in English, but also in French and we have French courses when we went for the first time there so the beginners could choose to have a French course for a couple of months. So I learned a little bit. I cannot say now they know enough to go didn't want to talk about difficult for me, like switch back to the English system. Like you were living in France or Geneva in France because it's close to the border, isn't it of France. Very close in the border is open so you can just pass.

So did you spend any time exploring around Geneva and parts of France that you're in or did you have any chance to see see see, see the country while you were there?

Raluca: Were a couple of citizens in Bern, I think from Switzerland, but it's my favourite city. It's a small city but it's very beautiful and I think when somebody has the chance to go to CERN, there are a lot of things to do and to visit. You can just go hiking, if you're bored because nature is around there. Yeah.

Are you do you ski or snowboard at all? Is that somebody a winter sports person as well?

Raluca: there was a ski club at CERN, but I didn't go this year. But now I will. I will try to go with them. Because at the beginning I was not a very winter sport person. But now I'm going to push myself because it seems very, very fun. Yeah. It can be quite expensive and expensive sport but when you're living close to the mountains, it's it becomes a lot more affordable. Because you know it's only a 30 minute drive away or whatever it is, whereas normally it involves hiring an expensive chalet and you know, and all that stuff, but I think we're living in a country. It's sort of it's it's a good thing to do. Yeah.

Okay, good. So you came back from 15 months, I suppose at CERN into your final year.

Raluca: So actually, before before I move on to that, actually, so what would you say was the most enjoyable or rewarding part of the placement and you mentioned that you're part of this international community and learning heaps of stuff. There's anything that really stands out for you, from what you've got from your year from your year at CERN. Yeah, so first, of course, is the whole community that I found there. So I just felt like my place was exactly that. Well, I think one of the most important thing is that I learned how to multitask a lot, how to be able to learn new things, because I think a couple of years ago, I thought that okay, if you if you feel like you

can now learn this specific skills, skill, you will never learn it. But in the end itself, there were so many things I had to learn that I didn't even have time to worry about not being able to find it and yeah, I learned details that I never thought I would understand. Also, not in my area of expertise. For example, details regarding cables, fibre optics, I never I never got in contact with that before. Yeah. And also, it wasn't my first experience seeing how what you're doing is actually used and has impact on people. And it gave me this sense of purpose, because I had full responsibility over my tasks. It made me even more excited to do a good job. Those were the most important thing is I enjoyed it.

21.4 Final year

Good. Okay. So yeah, so we, you came back from your year at CERN, or just over a year at CERN last September and came into your final year. So what were you doing your final year project? Can you tell us a little bit more about what you did for your final year project?

Raluca: Yes, I was working on a project related to neuromorphic computing, which is concentrating or building computer architectures that lay the working in the structure of the human brain and I worked on SpiNNaker, which is the computer built at our university. The idea behind it is to implement spiking neural networks, which are artificial neural networks that have computational elements that mimic real neurons. And my job was to build a new neural models implementing Spinnaker to be able to see if it provides the right behaviour that we were expecting. And I was lucky to be lucky able to live for it and in the end, it was working. And I saw how, how complex and complicated it is to hear even one neuron. How complicated is the field another full network of neurons. And the reason I chose this project is I believe this area of research is very interesting. Being able to simulate the real brain will not only bring progress into computer science, but also neuroscience, for example, just a platform where neuroscientists to analyse the brain and see how it works or how does it do that? Yeah, was a great journey for me this year. Yeah, so that was that's part of the Human Brain Project Isn't

it always amazes me the the, the size and complexity of the human brain is far beyond what we can currently even begin to simulate. I think I forgot the figures, but how many how many mice brains is spinnaker now up to how many how many mice can you can simulate the brain for the rat cortex at the moment Right.

Raluca: Yeah, but other than that, I guess it is evolving pretty fast.

So who knows? And that's a lot of that's a lot of silicon, isn't it?

How many processors do you have on Spinnaker at the moment? And it cause? I don't know for sure how many they are to be honest. So the part I think, was it last year or the year before they passed a million core. So a million cores which is which is an insane amount of power. But it can it can you say all that power. You can only simulate a rat Cortex A brain a bit of a rat and use a huge amount of power and your your brain the brain inside your head does that on something like about 40 watts or something? It's it's much much lower. It's amazing, but it's amazing. It works at all. With all the speed and learning progress of the brain. I think it's not even fully understood by neuroscientists. So simulating it it's even harder because it's not fully understood. Yeah, so your supervisor, your supervisor for that was

Right. Good. So so the next thing then is, well, there's two things really. You're planning for the future. So I know that sounds actually unusual in that a lot of students who go to CERN aren't recruited so that they can go back after they have done an a placement. Students go back to their employer after they graduate, but with a lot of CERN students, that's not the case because they have this educational thing where the money that they're getting from the European member states, and sort of condition of getting that money is that they educate and train people all over Europe. So it's not just about recruiting lots of engineers doing a placement there. It's about spreading the knowledge of all the technology and systems and know how that is it sends spreading that around as widely as possible around Europe. And so in your case, you are you are actually planning to go back I think you might be the first University of Manchester students go back other Josh Dawes, went back there to do a PhD. But there aren't so many students who go back not because it's not a nice place to work. It's just because the way that there's paper temple students ships are set up as is to encourage people to learn some stuff and then move on to somewhere else. So your plans for the future are to go could to go back to CERN. Is that right? Yeah.

Raluca; So luckily for me, I was offered a graduate position in the experimental physics department on our major experiment that they had there.

Is that a different group to what you're working on before?

Raluca: Yes, it's the experimental physics department and it's on the ALICE experiment, which I can talk about if you want. Yeah. So that'll be another adventure. Then I guess, in terms of stuff that your stuff that you're going to be learning

Do you know, do you know too much about the group or what they're doing it?

Raluca: Yeah, I already know something because I wanted to get

into that position. I knew about it. But of course, I was not sure I would be accepted. And I was. It's called a large iron collider experiment. It's one of the experiments at CERN where they studied the properties of matter, at extreme densities and temperatures, the collisions of the accelerator, and I will be working on a technology for organised distributed data analysis that is dedicated to storing and accessing the physics data output from ALICE. Yeah, and this is very important for the allosteric output to be able to be accessed by scientists and analysed.

So the Graduate Programme is on is it called a Graduate Programme and do they call it a scholarship or what? It is a fellowship or fellowship? Okay. So how long is the fellowship for is it for a fixed term?

Raluca: It's three years, three years, six months, and then you can extend it. It's a trial period of six months. And then if you manage to First up, you can extend it up to three, three years.

Do you know what you might do?

Raluca: Um, no, no, three years is obviously a long time.

So it's almost as long as you've been studying as an undergraduate to do you have any sort of plans after that or any sort of things that you're thinking of in the future that you might like to do after this period of three years?

Raluca: I'm not sure to be honest, I could. Because once you finish that you can also continue at certain with, for example, a subcontract if you want to, and he managed to get it, but I'm thinking that maybe I will find something that I'm very passionate about a specific kind of software technology. And who knows, maybe try to have a master or PhD if I'm not sure the moment or just stay there and work head on carry on being an engineer without doing any further study. So I guess I'm there's lots of options there for the future.

21.5 Advice to your former self

Sounds good. So I've got two two final questions, really. So one is, these interviews are aimed primarily at other students. So what advice would you give to students who are looking for either summer internships or placements during their first and second years of study? What would you what would you advise them to do? What do you think they should be doing?

Raluca: I think I could talk a lot about that. So yeah, first, I think they should try to find companies or organisations that they identify with what they appreciate. Because this way they will be more dedicated to make a good application. And also if they get

accepted, they will be more happy to be happier to work for that specific company or organisation we should try to find companies or organisations that they identify with what they appreciate because this way they will feel more dedicated to make a good application. And also if they get accepted, they will feel more happy. They will be happier to work for that specific company or organisation. Then a lot of effort, a lot of effort has to be put into the application. I know myself because in the second year I realised that a lot of talented students that want the same job, it's very important to be able to stand out with some personal touch of the application or opening the CV so putting a lot of effort in that is important. And then I think, I think it's just pretty much it.

I think the point you're saying about personalization is very important, because I hear stories about how many the volume of applications that the more competitive schemes and CERN's probably one of the more competitive ones is, you know, it's hundreds a day of applications. Yeah, you have to remember that there's a human being on the end of that, who has to somehow try and find that stuff that they're interested in. And it's the easiest way one easiest way to do that is to go through ones and just avoid ones that just look like they're generic applications.

Raluca: I'm sorry to say that I have a third advice. Do not apply to one company. Right. It's a very risky. It's very important to be able to have a backup, or who knows maybe that backup that you have will actually end up being the company you want to work for. You never know.

So what were your backups? I know, because it sounds obviously quite an attractive employer. Did you have any backups other than certain what there are other places you were thinking? Well, if I can't get certain then I'll do this. What other places were you considering at the time?

Raluca: Yes, yes, I had. I had two others. There was DataStax and Koderly right. So I got to know that I could go there.

I think it's always tricky balancing quantity of applications with quality because you can either do one or two really, really good high quality applications, but then if they get rejected, then you're stuck. And the other on the other end of the scale is the sort of scattergun approach where you just apply to lots. If you're doing lots of applications, then no, I've seen students do applications for 50, 60, 70 different internships or placements. And obviously, the more you start doing that, the quality starts going down. So you've got to try and balance the quality with the quantity and that's, that's not always an easy thing to do.

Raluca: They also have to keep track of the 70 applications.

You may remember what you said, why did why was it I wanted to apply for that company, what is it that they do? And I think that it can also be bewildering

as well because I mean, how did you find balancing those commitments with manage managing, you know, managing your second year of study, but then also you need to have a social life as well. And balancing all these commitments can be quite hard sometimes I think.

Raluca Yes, no, it was not easy because second year, it's busy with more things to do more labs and then you also want to apply for jobs. And you also want to have a social life. Exactly. So I try to organise my making pretty clear to know what am I doing today or tomorrow I, I usually make lists with what I have to do. And take, let's say, for example, that weekend that's free or at least one day in the weekend is free. And but I was very passionate and I really really wanted to get a placement. So I dedicated as much time as I could to them.

And it's good to see that paid off.

Raluca: yeah, definitely.

21.6 The future

Okay, so the other question I had was, so we talked about your plans for the future. So what do you what would you say the most interesting thing or things happening in computer science or technology at the moment there any things that you think this is, this is a growing area? This is something that's going to be really, really interesting in the future?

Raluca : Yeah, actually. I will answer it in a personal way. neuromorphic computing that I talked about before bit of above unbelievably hard, actually. Yeah, yeah. Okay, so that's that.

Is there any crossover there? So we've talked about physics and the CERN, the Large Hadron Collider and then you've talked about neuromorphic computing. And there are two, almost entirely separate fields. Is there any are there any sort of interesting crossovers between the two or any things that you think anything's you think might you might be able to apply from what you've learned in your final year project to to physics

Raluca: Yeah, this could be because, as we know, there's a lot of data generated from the physics experiments at CERN. We have around **1 billion collisions per second**. So we always have to find ways to store and process that data very fast. And very good not have those somewhere. So technologies such as neuromorphic computing, and this kind of architectures could maybe be able in the future to process information faster or better. Of course, now the link is not completely clear for me. I believe there could be some more way of using that knowledge to process the physics data generated.

Because I know a lot of the computational power of Spinnaker, this neuromorphic computer has yet to be used. So you can build a very powerful computer but then you need people to be able to understand how to operate and then use the resources that that such a machine provides. So yeah, that's that that sounds that sounds very interesting. Okay, so I think I think I've covered all the ground I wanted to, it's interesting finding out a little bit more about what you've been doing what during your time here. It was great to hear about your experiences of CERN, and what you've got planned for the future. Is there anything else you'd like to add before we close off?

Raluca: Nothing, just maybe good luck to the students that are about to be in a placement or find a job?

Yes, thank you for joining us Raluca. Now, I always struggle is I never know with the name. Who could do I emphasise the A or the you where do I put the stress? Is it RalUca? Or is it RAluca? Luca is Raluca right? Okay. Yeah. I, I because we have we have several other students called Raluca. And I always struggle with getting the emphasis rights

its a difficult name for somebody who's not Romanian.

Yes. I don't think there's no there's not a near equivalent. But thank you, Raluca for joining us,

21.7 Disclaimer

Coding Caution

Please note these transcripts are generated with speech to text software and are not perfect word-for-word transcriptions. Some speech disfluency has been manually removed and links, cross references and pictures have been manually added for clarification. Extra (non-quoted) words are given in brackets like this: (some text added afterwards).

Chapter 22

Brian

Meet Brian Yim Tam, shown in figure 22.1 he graduated with a BSc degree in Computer Science with Business & Management and Industrial Experience in 2022.

Listen to the episode by clicking Play below, or subscribing wherever you get your podcasts, see section 20.1. An annotated and edited transcript of the audio is shown below.

22.1 What's your story Brian?

A draught of the machine generated transcript is available but needs more editing.

22.2 Disclaimer

Coding Caution

Please note these transcripts are generated with speech to text software and are not perfect word-for-word transcriptions. Some speech disfluency has been manually removed and links, cross references and pictures have been manually added for clarification. Extra (non-quoted) words are given in brackets like this: (some text added afterwards).



Figure 22.1: Brian Yim Tam. Picture reused with permission from [linkedin.com/in/byt411](https://www.linkedin.com/in/byt411), thanks Brian.

Chapter 23

Jason

Meet Jason Ozuzu, shown in figure 23.1 he graduated with a BSc degree in Computer Science with Industrial Experience in 2022.

Listen to the episode by clicking Play below, or subscribing wherever you get your podcasts, see section 20.1. An annotated and edited transcript of the audio is shown below.

23.1 What's your story Jason?

So welcome Jason. Thank you for joining the show. And I wanted to talk to you a little bit about what your journey's been from before you started studying here to now four years later. There's a lot has happened in the last four years so tell us a little bit about how you came to study computer science. So can you tell us a little bit about yourself and where you from? Where you studied, where you studied before you came to Manchester?

Jason: Yeah, so yeah, London through and through born and raised in London, living in London my whole life. Ethnically, I'm Nigerian. So my parents were born and studied in Nigeria before moving over here later and in terms of locations in London, I'm from East London. So it's like my main stomping ground. I guess I'll talk about how I got into computer science.

Yeah, yeah.

Jason: I'd say the main way, computers are kind of is a weird one and I tell people the story, they don't really believe me, but I actually didn't really have much of an interest in computer science at school until I got to sixth form

Did you do it (computer science) as an A level?

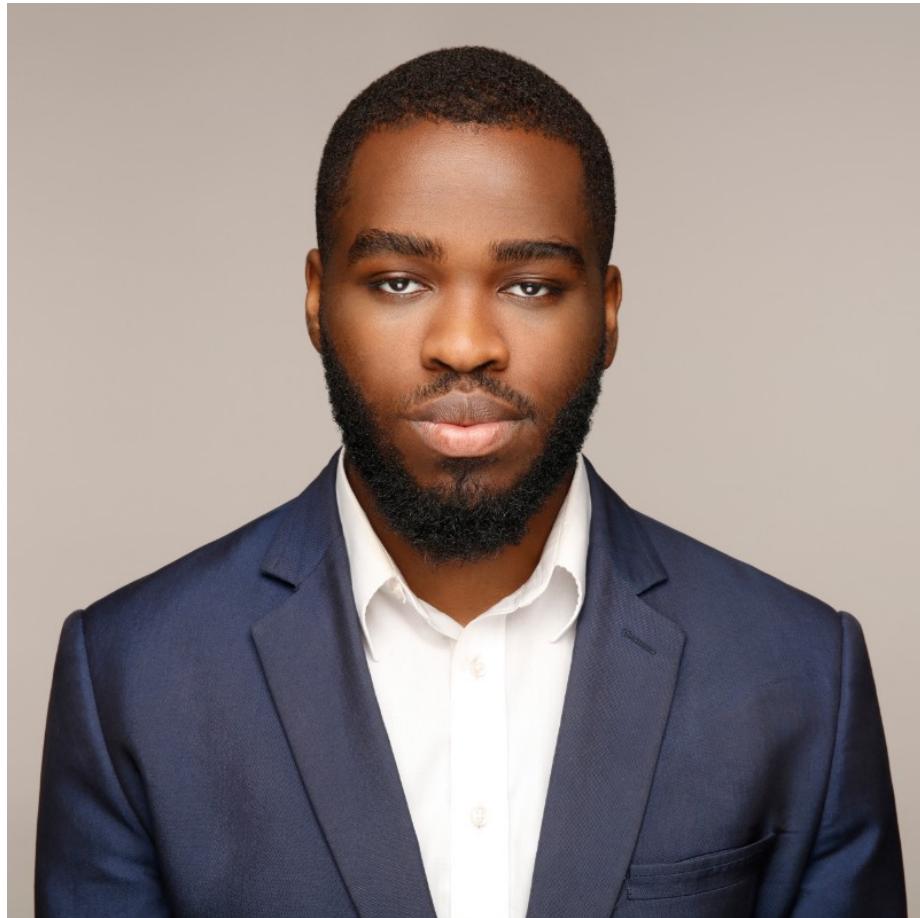


Figure 23.1: Jason Ozuzu. Picture reused with permission from linkedin.com/in/jason-ozuzu-a87049173, thanks Jason.

I'll get to that. That's part of the story. So I did, I did computer science and economics at GCSE those two that I chose. And essentially I really liked economics. I found it really interesting. And yeah, I wanted to take that further. I was kind of a **finance bro'** so to speak. I was really interested. I wanted to see where I could take it, but the problem was that when GCSEs were done and dusted, there were some issues with the teacher, our teacher, teaching us wrong parts of the spec and stuff like that. Essentially the grade I got in the final GCSE exam was lower than what I needed for the sixth form I would eventually be moving to off the back of those grades, if that makes sense. So my plan was kind of to get my GCSE and to move to like a like a more, a better sixth form. So I could get better education for A levels but the problem is that due to, you know, issues of the teaching and whatnot, the grade wasn't as high as it needed to be for me to take that subject at A level at the sixth form.

Right

Jason: So you know, originally I wanted to do physics, maths and economics, you know I was planning on going down the finance route. Yeah, but yeah, they wouldn't let me do it. So I thought okay what's the next best thing or something? I enjoy and something I found pretty you know, easy at GCSE: computer science. I found I liked it, I enjoyed it. I found it pretty straightforward. I got, I think I got one of the highest marks in my year, for computer science, at GCSE. So, you know, that's pretty confident, I could, you know, do it and, you know, it makes sense.

So, you did physics, maths and computer science?

Jason: Exactly. So I started that and yeah, that's really enjoy that. I grew to really love it. You know I got sided, you know, picking up getting prop learning how to code properly because, you know, the code you do at GCSE is very rudimentary and yeah, that kind of took me to university. I discovered computers. That's kind of came to me. It's kind of, by chance because if my sixth form had allowed me to do economics, I don't think I'd be here today. So yeah, it's pretty crazy, pretty crazy, but yeah,

So it's doing A level that made, you think, actually, this is an interesting subject I like to do. I'd like to take further than just A-level.

Jason: exactly

23.2 Morgan Stanley

Okay, interesting. So, and then so your story your arrived here, four years ago, you've done a three-year bachelor's degree with a year in industry. So can you tell us a little bit about that journey? I think you you did a placement at Morgan Stanley, wasn't it initially? You started looking for that beginning of your second year. So you tell us a little bit about how you found the job and that whole process.

Jason: Yeah. Essentially, just a ton of applications. I mean, I think the main thing for me, I wanted to keep my horizons pretty broad. I wanted to make sure I didn't limit myself to just tech companies or limit myself to any kind of niche. So and I always found finance pretty interesting how that mentioned before, you know, I always had that always had that prior finance interest. So essentially I was applying for you know, traditional tech companies, investment banks and yeah, after a bunch of applications I was able to get to the final round, the assessment centre of Morgan Stanley. I managed to smash that and then you know, got called in to do the placement. So that was yeah scheduled to be a year but obviously was caught a month short because of COVID. You know it's funny because I literally started in the middle of COVID.

So that was all online. I guess.

Jason: Exactly.

Were you in London or Manchester?

Jason: I'm back and forth. I mean coming up here, see some friends, do some work, why not? And yeah, so I only go to go into the office in the last, maybe two months of the placements, right? It was a bit, not ideal, but, I mean, you know, life gives you lemons, you make lemonade right? So, just took it for it was, and still managed have a really great experience off the back of it.

Did, you do a rotation? Did you do two six months (rotations)?

Jason: I spent six months with the equity derivatives team, so that's working on like, I mean, not to get too jargony but essentially derivatives are like a type of financial product and I was with the equity derivative team, so like traditional like stocks and options and what-not. So I was working on technology. That would essentially basically automate the process of passing requests for different financial products coming into the firm. So it was kind of like an NLP (Natural Language Processing) type project where clients will be able to kind of use chat system that Morgan Stanley had available and then some automated passing would be applied to their messages to basically determine like what product they want. Yeah. So that

was the first rotation and the second rotation was with the FX flow team.

FX is?

Jason: So that's foreign exchange for "forex". Forex essentially. So. Yeah, working like the currency, the currency exchange system systems and stuff like that. So yeah. I was able to learn, you know, a lot on both sides. I think the first one, the first rotation was kind of like a research. You type project what else kind of doing you know, some AI type stuff. Second rotation was more of a this traditional softeng role, I was kind of clearing out tickets. Just helping out of different bits and bobs, didn't really do anything, but specifically, I kind of just part of the software engineering team and this helped out and then I guess after my placement actually.

So back to, because this is quite interesting for now economics. So a lot of people who do software engineering placements in banks like Morgan Stanley, Investment banks, like Morgan Stanley won't necessarily have any financial knowledge. So, how useful was your economics?

Jason: GCSE?

Your general interest in it, as well. Some some engineers have little or no interest in economics. It's just well, this is a, this is a place where they do software engineering (and pay good salaries) so I'll work there. How useful is the economics interest in helping you with that?

Jason: It was really helpful because I already knew a lot of what they were talking about. At least the basic ideas of what is a stock? What is a bond? What is this? What is that? That made that a lot easier and then additions that I took *Fundamentals of Finance* in my in the second semester of university kind of in, not to the placement. So that allowed me to kind of top up on my knowledge and make sure I kind of knew that at least some of the basic concepts. So, but even once I started, people are really like happy to explain stuff to you and explain concepts to you. So one thing, one thing I really the point of doing was to schedule meetings with the finance side of, you know. So basically any people, the finance people would work with because kind of the way things work in Morgan Stanley is kind of the tech teams are separate from the finance teams but you usually have a couple points of contact where you can kind of, get clarification on, you know, the requirements of what you're building or something like that. So I was able to kind of talk to finance, people. Get a better idea of what was going on as well. Talk to the people on, on my side, on the tech side to you knew what's going on. I'm from there. You know, I think I think one thing I'd say to people is that I think even if you don't have much finance knowledge, it's less about what you know when you

start a more about what you're you know, willing to work. You're willing to put the work in.

Being able to understand the business is always beneficial. You're not just a techie to write code, in a back room somewhere. You need to understand the business requirements and what they mean and how a business people see it, or how the traders see it.

Jason: Morgan Stanley is quite good with that, I mean there's training the first week you do like group training and where you get like a bunch of different talks. So there's tech specific talks, we're like which are just going to take people's understand that the tech stack and stuff and then there's general finance talks which are basically just you know what is an investment bank? You know, what does that mean? What is the buy and sell side and whatever?

23.3 Google

So, following on from your placement, you've got a summer internship which was remote as well?

Jason: COVID, the gift that keeps on giving! So that was, I did some internship with Google which was actually yeah. It was a story that one. I've always been pretty ambitious. So for me like the placement I wasn't dead set amongst only obviously I did really find finance interesting but I wanted to hedge my bets and see, you know, after I think I finished a place with about three or four months to spare before I started university again. So I figured if I have that time and COVID is still a thing, I'm not gonna be going anywhere. So I might as well, you know, do an internship. So I was applying to, you know, a bunch of different places like Facebook, Bloomberg, a couple hedge funds. I was interviewing alongside the placement, which was pretty tough to be fair because preparing while working a full-time job. Really. Difficult a lot of late nights. Well, you know, I'd log off work and then turn around and be up, like, hackerrank (see section 10.2.5) and leetcode (see section 10.2.4). But yeah, like, we'll see. It worked in my favour.

Do you think you could have got that job at Google without the Morgan Stanley experience. Having that bit of experience helps because they'll ask you about situations: in my current job I'm doing this and that can help. Whereas, I think going from zero to Google is a big ask for a lot of people.

Jason: exactly. It definitely helps I'd say just from especially from a confidence perspective because by the time I'd even started applying and interviewing I had maybe five months under my belt with Morgan Stanley. So I was pretty confident in my abilities. I've gotten

great feedback from my managers and everyone I'd worked with, put up with me pretty well. So from a conference effective and also from like a this in terms of my actual technical ability, I would definitely a lot stronger than I would have been. **One of the biggest reasons I recommend doing a placement to people was it allows you to tie together all the different concepts you learn in the bachelors** you kind of you pick up all (these) different concepts, but they come at you so fast. You don't have time to stop and figure out where they will, where what they mean relative to each other. I was able to get really competent and that made into the interviewing process like pretty straightforward. I got to pretty much everywhere I applied to, but you know, Google ended up coming through first. So I was pretty happy with that, basically the best company I could ask for.

What were you doing at Google? I mean, I guess there's probably stuff you can't talk about but generally, what were you doing?

Jason: So yeah, I guess I was working with the Fitbit team. Fitbit, for those that might not know, was acquired by Google at the start of 2021. So that was literally a couple months before I started having they were acquired February or I think March 2021 and I started like July 2021. So right Fitbit hasn't actually been a part of Google for very long. It was like a Google internship but at the same time a Fitbit internships I kind of got a bit of both

interesting.

Jason: Yeah it's a bit weird because their systems hadn't been fully integrated with Google systems yet.

So migrating things

Jason: a lot of moving parts, lot of things going on. So I got to work with some Google people. Some people who don't know anything about Google or just with Fitbit, all they know is Fitbit, they are part of Google Now. So it was interesting and regards the actual technical details of the project I worked on, it's basically a smartwatch app so I was working on app for an upcoming Fitbit smartwatch. I can't go into details about what the app does but ...

Google watch!

Jason: essentially so that was that was really cool. Yeah. Got to I've never done it kind of like smartwatch app development or even like that before.

It's quite low level stuff was it? A bit of web application stuff but talking to to the hardware?

Jason: So basically the application star so designing that Google's really big on making sure your design is solid first before you start coding. So I couldn't even really do any coding for the first month of the internship. I did like a little bits here and there to get a feel. So did a lot of like app design stuff. Figuring out what it would look like on the smartwatch, what it should do in addition to some low-level stuff to actually do some new hardware. There were some new hardware functionality, that was coming to the watch that they were bringing bringing out that we wanted to make use of. But the problem was, there were no existing interfaces, like API interfaces between the code and the hardware for that particular piece of functionality. So, on top of having to design the app I also had to design that API from scratch. Thankfully I had access to the source code for the Fitbit SDK like at development SDK. I was able to dig into that and grab what I needed and glue stuff together and figure out with help from my manager. Thankfully the project was good because I got a feel for both the app design stuff and also the low level. How do I actually make this work from a hardware perspective? What should the API look like and I was left to do that by myself.

Good experience!

Jason: It was really good experience.

23.4 Final year

So when you come back to university, after your two placements and internships, you came back to do your final year here and you essentially trying to pull all things all together. So what did you do? Your final year project on what?

Jason My final year project was a hand, gesture recognition system. It was an exploration the feasibility of making a really portable easy to use hand gesture recognition (system) that can use any machine, any architecture whatever. It evolved to taking a lot of existing technologies and bringing them all together. And yeah, one of the technologies I use is actually developed by Google called media pipe which I made use of to perform the hand tracking and feature extraction, which I then use a neural network to classify

So taking some sort of video feed?

Jason: Exactly. So essentially a video feed being passed for if hands present inside the view of the camera features like strategies are basically just points on the hand and then that's fed into a neural network, which basically tells you okay, what gestures being performed. The idea is that I would create both create a really robust

and lightweight system that can be used anywhere and also use it and apply it to a particular use case that improves the user experience, both from disabled and able-bodied people. So I took that and then used it with Spotify to make a music player which has allow you to increase and decrease volume. That was that essentially what I did,

Good and so again, it gave you a chance to dig into a bit more AI stuff you haven't done a huge amount of up until that point?

Yeah, I'd done a bit on the course but I wanted to see what I could do to make AI work for me and just do my own thing with it, be creative with it because nothing like this had really been done before, especially using the technologies I was using. So that was the most interesting aspect of it. So I'm bringing all the different pieces together and seeing what if you could actually improve these experience for something which, yeah, which was proven to be the case, with some user surveys. I did with able-bodied and disabled people. I was pretty pleased about it.

That's all done. You've literally just come from your presentation. So that's the end, your last thing, your last formal thing before graduation. So and what are your plans for the future. What? What comes next? What comes after?

Jason So the plan is, thankfully, I've received a return offer for both Morgan Stanley and Google, but I went with Google just because it aligns more with what I want to do long-term.

In London?

Jason: Yes, so I'll be starting with Google in October later this year.

Congratulations!

Jason: thank you so much. I'm just gonna take the summer easy, gonna be travelling, taking it easy a bit before after, you know, enter the working world.

It's quite a rare opportunity to get that much time off when once you're working, you don't get three months off do you!

Jason: exactly

Enjoy it!

Jason: All my placement money is going on this essentially. Longer term than that, I'd still find finance quite interesting. I think a part of me will always be interested in doing something that blends both tech and finance

I was looking at Apple, who have been in the headlines recently because their pay system. They are behaving more like a bank. (Bradshaw et al., 2022)

Jason: Oh yeah, yeah, expansions to Apple Pay.

They've been partnered with Goldman Sachs, But now they're doing everything on their own. That's an interesting space what's happening there. Who knows what the future will look like? What is a bank going to look like in the future?

Jason: Yeah, exactly that. **Everything's just gonna be one big tech company!** Maybe I should move to finance in a couple years. I love to you know maybe because like maybe a hedge fund, the work on like tech stuff there or go to kind of something more small to medium size. Like some kind of like fintech startup that kind of thing. Maybe that'd be cool. Like maybe three, four years from now. For the time being starting my career with Google and I'll see how well that goes

Well enjoy and good luck.

Jason: So thank you.

23.5 Minority report

To finish up then, this bit is called *Minority Report* so I'd be interested as you are a member of a minority group, you're black. I'd be interested to hear about what your experiences have been of being black in computer science but being black in the university but then also in the workplace as well. What's that been like, honestly? What could the university do? And what could we perhaps we in computer science do to make it more welcoming. It's difficult being in a minority for lots of different reasons. And then, perhaps what employers can do as well? Perhaps you could say a little bit about, Morgan Stanley and Google are both, keen to improve diversity in their workforce, but it's it's an uphill battle for everyone. Any thoughts on that and what how we can do it better in the future?

Jason: I think it's more just a case of my experience has been good. Like I've never felt, you know, out of place. I made plenty of friends on the course. I think the main thing is just there are very few black people in my yeargroup. I've been one of the first thing.

How many?

Jason: I think when I first started in first year, my placement has shifted me away from some people. I think maybe about three or four people total

Out of a total of 250 or so?

Jason: Yeah, similar at Morgan Stanley

Was there more diversity?

There was but not much. I think in general I'm pretty used to being the only black person in the room or being close enough to it. I mean there were some other like a black people at the company,, I was able to get meetings on the board with but oh but particularly in in the tech space not that many having only only a few but I've made a lot of new contacts and that were black. In general, I'd say the problem isn't necessarily how I mean, bloody won't treat anything different. I think, especially in our part of the world, we're pretty far past that like nobody's, I think until I have never experienced any kind of prejudice, like while I was at work or anything like that but I'd say the main issue is just getting more people, more black people into that. These kind of spaces. So, like, especially considering that from like a socio-economic perspective people from my background don't tend to go to higher education or to even study computer science and to get into tech. So, I say the biggest thing, you know, both campuses and companies can do just keep them when they're doing. Keep trying to get more black people into these spaces and from a university respective, I'd say more could be done for regardless of that kind of helping with the financial aspects of university, offering more scholarships and bursaries and this, and that just kind of doing more to kind of ease the transition that makes sense.

Yeah. Yeah

Jason it can be a financial problem for a lot of people

Because I think computer science, regardless, whatever your gender or race or whatever. I thin computer science, sometimes people don't feel like they belong, right? So it's kind of making people feel like you belong here, whoever you are, and whatever your background, whatever you're interested in. Computer science is a subject that is for everyone. And it doesn't quite have that appeal or maybe no subject does really truly, but making it more appealing to a wider group of people would be beneficial for everyone. Difficult to get there.

Jason: Yeah, exactly. But yeah, in general, I'd say that experience has been I think in university in general, think the UoM I think the stats are like, less than 3% of the student population is like black or anything like that.

So the proportion in computer science pretty much reflects that.

Jason Yeah, university. Exactly. This aren't many black people in.

The thing that really hit me was when you were in Morgan Stanley and you posted something on LinkedIn (Ozuzu, 2020) with your picture in Times Square (New York, see figure 23.2).

I've visited investment banks and that was when it kind of hit me. I hadn't realised, how white it was until I saw your face and thought, oh my god, that's Jason!

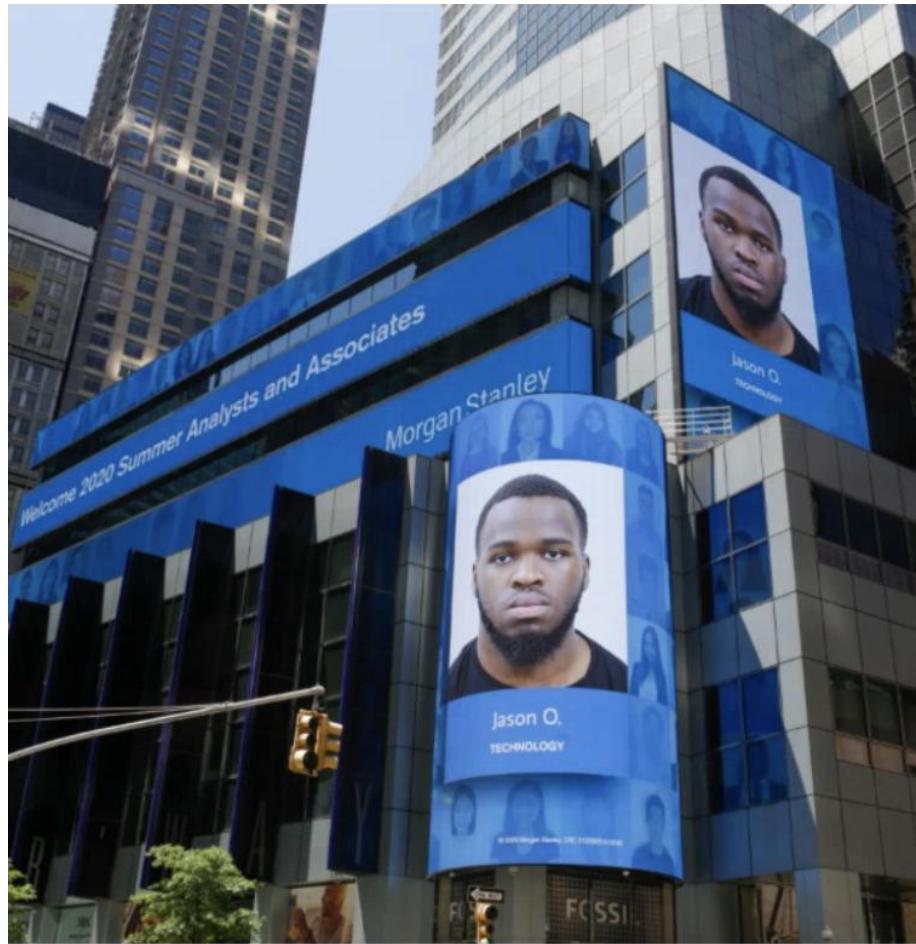


Figure 23.2: Jason's portrait being projected in Times Square New York, during the pandemic in 2020. Picture re-used with permission from Jason. As of 2022, this post has 17,712 reactions and 713 Comments on LinkedIn. (Ozuzu, 2020)

Jason: Ha ha

Oh, I recognise him. How white that world is, its a very privileged world, there's lots of money but it's also very very white.

Jason: Exactly. And I think, I think that could hold some people from, like my kind of background back, people might grow up in like pretty black areas. A lot of people, their schools, all kind of look like them. And then they kind of shunted into the professional environment and they're not even, they haven't really had many interactions right people in general. Yeah. So that can be that can be issue for some people as well. Like thankfully I went to a pretty like diverse school. I got a pretty good mix of people at my school. I know people who like, in London they went to schools which would maybe night, 80 or 90% black. So it's kind of like when they do kind of come to university and kind of leave that bubble, it's they can feel bit out of place. Yeah, that can definitely be an issue.

I know in the US they have universities that are much more traditionally black, (see Historically black colleges and universities (HBCUs)) and we don't again really have that is kind of yeah, strange on that.

Jason: Yeah. But I'm a big believer of different people mixing together and like, kind of, yeah. So I mean, in my opinion, I don't even think we'd even really need put like black campuses or anything like that. I think the biggest thing is just, just adding more diversity. You're getting a bigger mix of people and yeah. Yeah. But that's obviously something that has to be done over the very long.

Yeah. Part of it is role models, as well, people can see. Oh, you know. This is possible. Something people can then. Did you do any work with schools? Well, I know Morgan Stanley does quite a lot of work with schools. Did you do any? They do stuff in Tower Hamlets

Jason: I would have if it was in person but COVID there wasn't really much, I can't do. I was eager to, but I think when I go back to London, one thing I want to do is kind of start going to kind of like in the city schools and show. Like I'm just a regular guy from London. Regular black guy from London, like, nothing that crazy about me. I just worked pretty hard and managed to make something of this whole tech thing. So, yeah, I recommend all my friends, I'm always telling my friends, listen look at tech, tech is pretty good , pretty chill, you know, good money, you know, good work life balance. I would tell people, but I'm I think in addition to the racial thing, I say a briefing, the whole feel back from tech. And general. They kind of like what he is. It's not an easy subject. It's not easy thing to get into, but I think people think it's like impossible to get into you.

You have to be a genius programmer!

Jason: I tell people. Oh yeah, I'm starting with Google. You must be amazing. You must be a genius. NO I'm not I'm not, I work pretty hard and you know I'm competent. I know what I'm doing but I'm not crazy. I'm not autistic. You know, specialists of anything really? I'm just find it interesting. I've worked hard and acquired skill that was good enough to know work at these companies. So that's what I tell people, you know? It's just, it's okay. If you take your time, you're willing to be consistent and you know do it as like a proper thing. **I think a big thing of tech is that you kind of have to do it over long period of time and just gradually soaking all the skills.**

One of my colleagues (Ulrike Sattler) talks about learning, it's like sedimentation. It's like things settle slowly. You're not just going to get it straight away. It might take months, maybe even years to get it. For some subjects to actually click.

Jason: Exactly. What I think, a lot of people that I've spoken to try and get into tech and try and do coding. They could they do code for maybe like maybe two three weeks and then like yeah, I couldn't figure out. All right. You kind of need to do. Maybe I think you can't even say you've really given it a try until maybe six months. I say, three to six months of actual of graft and then maybe you can now, say, okay? Maybe this is or isn't for me.

23.6 One tune

Okay good. So to finish up then, this podcast is part of *coding your future* and it has a soundtrack (see chapter 17) with it. There are musical references in the text but because I'm older, a lot of the references are old. So, can you recommend us

1. One tune to rejuvenate our playlist with
2. One podcast that you've listened to or enjoyed
3. One film if there's time as well

So let's start with one tune

Jason: One tune.

It could be something that you helps you relax. It might make you feel happy or remind you of home or, you know, might be something that helps you study. Or maybe opposite. Something that helps you get your mind off study when you're when you're stressed.

Jason: I mean, if you're a big fan of hip hop, I'm a really big fan of hip hop, but that kind of thing, American rap and stuff. So, I

say anything by Don Toliver. (Toliver et al., 2019) People probably might not have hard of it. I'm big. Big, big, big fan, right? So anything by him. If you want, like a chill, like hip hop, kind of vibe type thing.

23.7 One podcast

Yeah, good. Okay. And then one podcast that you listen to recommend

Jason: one, I really recommend is under the skin, by Russell Brand. (Brand, 2022) So for those of you may not know that Russell Brand was a really big Reality TV guy for a while but went on a bit of a bender and lost himself to drugs and addiction and stuff like that. Then he rediscovered himself, completely reformed himself and he has a podcast called under the skin where he brings different people on to talk about different things. Sometimes he'll talk about spiritual things to talk about, the concept of the soul, and what it means like, life and death and stuff like that. You can get really deep like that. Or he might do stuff about social economic stuff and current issues and current problems we have in our society right now. And yeah, it's really good, like blend of like just a bunch of different things. So yeah, I recommend if anyone looking to kind of broaden their horizons,

23.8 One film

Sounds good. Well I'll put a link in the show notes and then the last one was a film any films? Any films that you'd recommend people go and watch. Netflix or wherever people consume films these days.

Jason: I recommend my favourite movie, probably of all time is The Matrix, the first one, (Wachowski and Wachowski, 1999) not the fourth one. That's come out now. Don't watch it. It's trash. The graphics are amazing. The CGI is amazing. The story. Yeah. Not so much I wasn't a great fan. But, I mean, the first original, the first matrix was one of the movies, I always tell people that and Iron Man (Favreau, 2008) are the things that made me like computers in first place. So I'd recommend The Matrix, (see figure 23.3)

How old is that film now?

Jason: That must be came out in the 2000s. Oh, 1999. 20 years ago Yeah, it's crazy you. Yeah, pretty much pretty much. Yeah, but yeah, good.

It depicts a dystopian future in which humanity is unknowingly trapped inside a simulated reality, the Matrix, which intelligent machines have created to distract humans while using their bodies as an energy source

The Matrix



Figure 23.3: *The Matrix* depicts a dystopian future in which humanity is unknowingly trapped inside a simulated reality, the Matrix, which intelligent machines have created to distract humans while using their bodies as an energy source. (Wachowski and Wachowski, 1999)

23.9 Advice to your former self

All right. Well, thank you, Jason and is there any other closing comments before we finish? We have one here that was advice to your former self. So what would you tell your former self or a current first or second year student who is studying computer science. What should they focus on?

Jason: I would say, the biggest thing is right, don't sell yourself short. You don't need to be a genius to work any of these places, why I've been telling people years below. You ask me about my experiences like you don't need to be crazy. Just work hard and know what you're talking about. Just be competent and believe. Honestly, have confidence, I think a big part of it is confidence. People kind of projects this idea of like, oh, I'm just not good enough to work at these places. And, you know, when people see me, I'm pretty outgoing pretty, like friendly guy. And they see me like, okay, I'm doing all the Morgan Stanley and Google, they think it kind of spins them a bit. They kind of think that in order to make anything out yourself you need to give up life and do that make that your whole life. I mean, you have to work hard but I still go out on the weekend. I'll still. I've got holidays lined up for the summer. I'm still doing my thing. So I think biggest, biggest piece of advice is have confidence and, you only and **you only need one YES**

Okay. That's a good note to finish on. So thank you. Jason

23.10 Disclaimer

Coding Caution

Please note these transcripts are generated with speech to text software and are not perfect word-for-word transcriptions. Some speech disfluency has been manually removed and links, cross references and pictures have been manually added for clarification. Extra (non-quoted) words are given in brackets like this: (some text added afterwards).

Chapter 24

Carmen

Meet Carmen Faura Práxedes, shown in see figure 24.1 she graduated with a BSc degree in Software Engineering with Industrial Experience in 2022.

Listen to the episode by clicking Play below, or subscribing wherever you get your podcasts, see section 20.1. An annotated and edited transcript of the audio is shown below.

24.1 What's your story Carmen?

So, welcome Carmen thank you for coming on the show. The purpose of this interview is to catch up with you, find out a little bit more about your journey from arriving here, four years ago through to your journey while you've been here and what comes next? So what should we start off with? Can you tell us a little bit about yourself and where you come from? As much or as little as you want to say?

Carmen: I'm Carmen, I'm originally from Spain, born and raised there and from small southern town Marbella. So, very different to Manchester.

And you went to school in Spain?

Carmen: Yeah, I went to school in Spain. Never moved abroad until I was 18 when I came to University in Manchester.

So what made you choose computer science as a subject?

Carmen: So when I was actually applying for degrees, I didn't know what I wanted to do and it was December and obviously applications closed in January and I had to start writing my personal statement, to a point where the head (teacher), a schoolmaster called me and it was like coming. Where's your personal statement? You need to



Figure 24.1: Carmen Faura Práxedes. Picture reused from linkedin.com/in/carmen-faura with permission, thanks Carmen.

start applying. And I don't know what I want to do. I knew I liked ICT and art and maths and then he was like, so have you thought about double degree in math and physics? I'm like, no. Physics it's not for me. So, then he said, oh, what about software, engineering, computer science, and I was like, what is that? So back home, it's not really a thing, it's still the idea that technology is what you do in a basement, antisocial. That's the stereotype. So then I was like, sure, let's just go for it. And then I ended up applying for software engineering and that's how I got here.

That's good advice. I think one of the interesting things is is a lot of decisions people make about what to study at university comes from family, teachers and peers etc and sometimes those people good advice, sometimes they don't give good advice at all!

Carmen: I told one of my family members that I was going to study software engineering and they had a full-on go at me, I feel why would I do that? I'm gonna have not a pretty life. I'm gonna be so miserable. I when I came here, realised, that was not gonna be the right thing was gender.

We'll talk about this later was it perceived as that's not subject for women? Or was that not not on the table?

Carmen:: I think at that point it was more, it wasn't one of the core traditional paths back home, more like architect, doctors, teaching.

Yeah, good. So, can you tell us a bit then about your story. So you arrived here in Manchester and you've been studying. So you actually I suppose fast forward to second year. What happened in second year when you you I think you decided to do a placement from the beginning from the get-go. Was it something? You thought your peers persuaded you? It was a good idea?

Definitely the latter. I came into Manchester not knowing what a placement was, but everyone kind of just said, do a placement, do an internship. It's great. So, I kind of got pushed into it changed into that degree. So added that onto my degree, and then started looking for a placement. Second year, was tough time managing that and like, dealing with rejections and all, that is a horny thing to go through. It's very draining very heartbreakingly but it's very rewarding at the end

Time-consuming as well? So how long did it take? How many roughly? How many applications did you do?

Carmen: And I was quite lucky that sense. I think I only did about 14 right? Oh with like maybe four of those being emails that sent out to startups because at the beginning, I wanted to go into more startuppy vibes. But then I decided to go through normal job applications, apply to about 10 and obviously you get the bit but the

imposter syndrome. When you get all your friends getting applying to the big companies, I didn't want to work for a big company back then. So they obviously get the job opposite first and you feel like you're not good enough or you're not making it but then the offers starts to come in and then it feels a lot nicer.

And I guess did you have many interviews before? The one you got? Did you have many unsuccessful interviews? Because it's not like an interview for university, some universities don't have interviews at all, so that's a whole new experience in a way, isn't it?

Carmen: Yes. Absolutely. I remember the first one I did actually was broke company. I didn't really want to apply for, so that was good. But even then, that went absolutely shambles. I we got a tour of the headquarters as well, and the factory. The point at which I was mansplained was very big. I walked out thinking, I don't care if I get the offer because I'm not coming here. So it was a big eye-opening time and then after that I was way pickier with the companies I applied for.

24.2 The Walt Disney Company

You actually did a placement and internship. Did you which one which way round did you do it?

Carmen: I did first my placement, the placement was at The Walt Disney Company (Práxedes, 2022)

What were you doing? Brief description of what you were doing for Walt Disney.

Carmen: So we're just an internal applications kind of improving efficiency of finance and HR teams across the globe.

And you were in one team for the whole 12 months duration?

Carmen: Yes for nine months

And your work I guess you're working with a combination of technical and non-technical people in that role in that year?

Carmen: Yes, it was mainly technical. It was maybe like one or two that were non-technical but that's about it.

So on the back of that then you've got did a summer internship after you've done a placement. I guess was getting interviews a little bit easier. Once you have the first thing under your belt

getting that first job is definitely the hardest and I feel like doing the summer internship after really helped get the most out of it

because having already know how to get to new code base and just get familiar enough to speed as much faster.

24.3 McKinsey

The summer internship was at McKinsey? What year was this this? When did the pandemic hit 2020? So you'd started your placement.

Carmen: Yes. So physically, no, I did it fully, right? You're right, you started remotely. I did my full placement remote and then I realised. That's not something I ever want to do, again, anyone does. And then when I got the office first summer internship, I got the ones. I boiled it down to was one that was remote and what that was in person and I went for the one in person, right? Because I cannot do another six eight weeks of remote and then McKinsey put me in person and we got to travel with the company and that was the best experience ever.

So that was McKinsey (see figure 24.2). So you were based in London, with different clients I guess?

James Oscar McKinsey (June 4, 1889 – November 30, 1937) was an American accountant, management consultant, professor of accounting at the University of Chicago, and founder of McKinsey & Company

James O. McKinsey

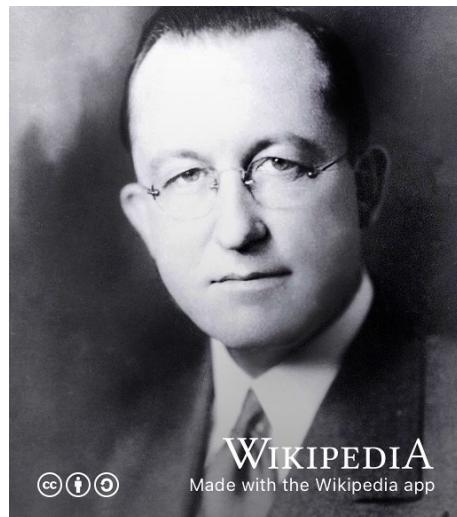


Figure 24.2: McKinsey & Company is a global management consulting firm founded in 1926 by University of Chicago professor James O. McKinsey, that offers professional services to corporations, governments, and other organisations. CC BY-SA portrait of James McKinsey by McKinsey & Company on Wikimedia Commons w.wiki/5ZTE

Carmen: So I was put on our US client but to get to the US during COVID you couldn't just get that easily. So we had to do two weeks

in Mexico first. So we travelled there. The day before we were meant to leave for the US, Mexico was on the red list. So I had to fly home and then never got to go to the client, but I got two weeks in Mexico. My first in-person job was two weeks in Mexico. I think that was pretty good.

I hear that Mexican Spanish is very different from Spanish Spanish.

Carmen: I couldn't understand the menu when I went to order food.
I was like, I don't know any of these

Someone was telling me the other day, Spanish person telling me that Google Translate when it does translate from English to Spanish, It does English to Latin or Hispanic Spanish rather than European Spanish Spanish. So, Spanish teachers here can always tell when their students are using Google Translate because it comes up with all these weird Latin American sayings and things that you would never say (in Spain).

Carmen: That makes so much sense actually. Yeah, no. It's you can really tell the difference. Especially there are a lot more formal than we are probably.

So you did your summer internship in London and then you came back to Manchester to do your final year?

Carmen: Yes.

So you did a project, what was your project?

Carmen: It was AI face ageing. So I could I actually use your picture as well for it and I made you look about 50 years old!

Oh no! Anyway, my face is getting older. So you kind of what you developing algorithms for doing that effectively.

Carmen: I used two different models on how to create that image? So that should help like, for example, missing people, that could help the police, it has different applications

What was that awful Facebook thing, was it face age or something?

Carmen: FaceApp

Robbing people's data...

Carmen: Yes. Actually while it was yeah, using pictures on them the Russian app. Yes, I actually include that my report because yeah, that was one thing we went we obviously shouldn't do. So yeah.

So, I suppose, one interesting thing is like, how did you find that transition from, so it's all about change, isn't it? You sort of you started as an undergraduate and that's what one change. And then you go to a completely different world of

work and that's another change and then you come back to university and it's another change again. So, how did you find that transition back into university?

Carmen: Going back to uni was a lot better than when I started Uni, because I felt like, when I started uni, I was a lost puppy. I didn't have any software engineering background, I didn't know anything about coding. I didn't know what Linux was. I was completely lost. Starting the job you also still feel a bit of like, what's going on. It's a new company. You're still big loss. Then when I went back to uni, because I already knew what it was like and I felt like during my placement, my confidence actually um, got built up a bit given that I actually managed to put to test my skills and managed to kind of just understand everything that's going on. I felt a bit more confident myself when addressing the courseworks and labs

And you were a (Peer Assisted Study Sessions)PASS leader?

Carmen: I was PASS leader in second year and past coordinator in third year, right? Okay.

So I guess that's kind of that's kind of fun. I think. Yes sociable and especially after the pandemic, having everything online.

Carmen: I definitely things physically definitely took to an extreme. The fact that in third year they say that yours have a bit more free time and just decided to sign up to as many as extra curriculars as possible and take on a part-time job, which was a bit too much but it was fun and kept you entertained.

Yeah, good. Okay. So, So what else have we got here? So it's got a questions about challenging things, about your job, what the most challenging things about your work life, would you say in terms of adapting to that and then working? Because working as a software engineer in a university or learning about software engineering university is quite different from actually when we try and simulate the real world and yeah we can do some simulation but there's an awful lot of simulation that can't be done. So what was what was you say was? Most tricky thing about adapting to the real world?

Carmen: I think the main thing for me was that I was used to seeing the things, but in a smaller scale and I didn't see it fully applied to the real world. So when I saw it at work or the little things that we had been taught uni kind of come together, it was it took me like maybe a couple months to just kind of put everything together and be like oh this now makes sense or this was taught this way because of this. So on bringing all that knowledge together was definitely a thing.

And pulling pulling together things not just from one module but I guess from several absolutely also seeing hopefully seeing a bigger picture.

Carmen: Yes, the it was that combination of everything that really set in

You finished your exams?

Yes. Last week two weeks ago, was it first of June?

You've had you've been able to enjoy the nice Mediterranean weather we're having!

Carmen: Yes it feels like home.

And, and what comes next? So what what your plans for the future after graduation?

Carmen: So I'm still carrying on the part-time job with the startup that I did throughout the year and they've offered me to go back full-time whenever I want basically, but then I also have the return offer from McKinsey, which I would be starting in September. Oh, so currently, I'm, I'm gonna move to London. That's one thing that I know for sure. And then in July, I'm gonna take that as a month of kind of test out what would be like full-time at the startup. Then I'm working for and then once I've done that, decided of, if I'd stay with the startup gift to McKinsey,

It's good to I suppose have that option of testing it, you don't always have that luxury.

Carmen: No, it's actually been very lucky in that sense but I do know that I love travelling and I know with McKinsey it's so client-based. I really enjoy that.

Although there's less travel these days with consultancies in the, you know, previously you'd spend you'd be like a road warrior, being on the road all the time and now, yeah, you still have to do that face-to-face stuff. But here's a lot of stuff via teams and zoom and whatnot.

Carmen: Yes. Um, which is also what I was scared about if that would affect anything. So I was speaking with firm themselves and they were like, there's a lot of people that also don't want to travel as well, So they would kind of allocate the clients. Hopefully, I'd be like, yeah, I want to travel so hopefully, I mean that I'd get to be with those clients

And I think while you're young as well. So while you're young, it's good to travel because you don't have any commitments or have less commitments and you enjoy it. Enjoy that kind of lifestyle while you can because it sometimes becomes harder as you get older. And so, the other thing is, so that was telling us about your story. The other bit is then minority report. So you are a member of minority group. That's grand. As a woman, what's you experience been in the university. I was going to look at the figures, but I think was something

around about 25% of the undergraduate population here (in computer science) are female.

Carmen: yes

So it not a disaster, but could be a whole lot better. That probably reflects what you saw in industry as well?

Carmen: I would guess. Yeah, I think slightly less from what I saw

Right?

Carmen: So I'd say it's pretty good in Manchester that we have that. It's definitely a thing that every girl knows every other girl in the degree because when you walk in, kind of it's a bit thing and when it's a girl walking in rather than a guy. I didn't notice it much at first. I didn't think I realised because of the lack of knowledge. I had microaggressions, that actually arise.

Was it like this at school? Sometimes you find that maths is, I think math is, it varies with countries, but I think in the UK, there are roughly equal numbers of male and female students do maths A-level, I think.

Carmen: We had, we were three girls in a class of 10 to 15, right?

So, it wasn't that much. And yes, that definitely did have an impact

And you've done physics as well, I know physics has a similar (problem).

Carmen: Yeah, it was like three to ten,

Right. Okay. So what could the university and employers do, to make campuses and workplaces more welcoming to women as a profession and as, you know, as students,

Carmen: I think for me, it's raising awareness on females and non-females. A lot of people identify people who don't identify female because for those who do identify female, we might receive these comments and we just take them in. We're not good enough or take them personally, whereas actually it has nothing to do with us or like if someone's underestimating my ability it doesn't mean it's because I have to prove them wrong. I have to do this, which is something that I constantly feel like, sometimes it's like, oh, I need to prove them just because I'm a female. But then also with people who don't identify as female to be understand. You can't mansplain real numbers to someone who's that, you know which has happened. And it's like, I'm 22 at University. I think I know what real numbers are. Yeah. Honest.

But part of the problem is the people mansplaining don't realise. So, that is not (always) malicious. It's just ignorance in a way, I suppose unconscious bias or whatever you want to call it.

Carmen: Yeah. So we had actually organised a few women and computer science events in Kilburn with another few students. And it was great because they actually made us more aware of other situations that maybe not, we can now reflect enough. Actually, that was not okay that if someone did that to me, and we've had conversation with people who actually try to mansplain to us like, oh yeah, mansplaining is not good and they'd be like, oh yeah, like I don't

They're mansplaning mansplaining to you!

Carmen: now for they'd be like, oh yeah, I don't understand why anyone would mansplain. And we just be thinking like you've tried to mansplain to us. So it definitely shows how more awareness needs to be raised in that regard.

The other thing is, making the subject (Computer Science) more appealing to women. There's lots of ideas about how to make it more appealing to women to apply to think about doing at university. Is there anything in the way that the subject is sold? Sold is not right word, in your case it was your head teacher. Who said, this is the thing. But some people look at it and think that's not for me because computer science is just about making lots of money, or it's just about building things that make other people lots of money. So one of the ideas out there is that you make the subject more appealing by showing examples of where it's beneficial to society, rather than just, you know, commercially important. Say for example, I don't know if that would have made a difference?

Carmen: I think for me just knowing what it was beforehand, I feel like the lack of exposure was a main thing. I guess this is still that stereotype that yeah, it's just maybe also gaming and not very inviting. So getting rid of that stereotype would definitely be more beneficial for me if I back in the day and

From an employer point of view, I guess both of your employers would have been welcoming to women, but I guess you have you kind of different problems in the workplace. But what do you think employers could do to make this as profession more appealing or more welcoming to women.

Carmen: definitely as well.I guess it just boils back to the raising awareness. Again, I've had work meetings where a developer, I was the only female in that work meeting and male developer asked for everyone's opinion individually on something. Excluding me. So it's just kind of probably raising that way and stuff. Like why what's the need to exclude someone or what's the, why do you need to make comments like that?

I feel like that's already quite inviting because it's quite difficult for you to call that?

Carmen: Exactly. Yeah. I guess it's your manager's job. I suppose

or whoever's chairing the meeting to say let everyone speak or you haven't asked Carmen for example what exactly and then it rated those uncomfortable situation it's like how do you deal with that? Especially in the workplace, where, you know, you have to be a bit more formal and ethical.

24.4 One tune

To finish up then I've got we've got a few things here. So, and this is one tune, one podcast, One film. Yes. So, we have a playlist that's associated with this podcast, Unfortunately, it's full of lots of dad music because its put together by me. I've been asking students. So, is there a tune that you can use to rejuvenate, or freshen up our playlist? So, it might be something that's important to you. It might be something, it help a piece of music that helps you relax or might even help you work, or take your mind off work or remind your friends family, and that kind of thing. So, any any one, tune that you would recommend?

Carmen: There's one that I've kind of got into lately, which is LOUD by Sofia Carson, (Carson, 2022) which is all about. How how women are starting to give a bit more voice and things? And that's making some people not feel uncomfortable. How do you spell? C.A.R.S.O.N. Okay. So that and it's like why should you feel uncomfortable? Now that I actually speak up so I thought found that quite motivational in times where it gets a bit tough.

24.5 One podcast

Yeah. Okay good. The one is so one podcast, I mean, you may or may not listen to podcasts?

Carmen: There's only one to listen to, one of them is my friends, it's called **ME time** (Maya and Emma, 2021). So I used to listen to that on the way to work throughout and it's just about real life, stuff. They speak about how to deal with things that you need had to do things in life. How to grow up as very related with this point in my life.

24.6 One film

And the last one is one film. It could be something contemporary. Could be something as just out now or that you seem in the last couple of weeks or it could be something older.

Carmen: You the last one I've seen is, She's the Man. (Fickman, 2006) Because a friend mentioned earlier they and it's quite ahead

of its time. It's like from 2000 and it's about how the females could also play football, right? And they didn't have to not be allowed on the team because they're a female. (see figure 24.3)

She's the Man is a 2006 American romantic comedy teen sports film directed by Andy Fickman and starring Amanda Bynes, Channing Tatum, Laura Ramsey, Vinnie Jones, and David Cross

She's the Man

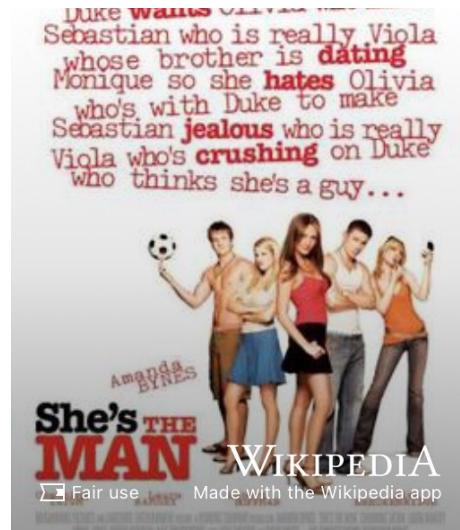


Figure 24.3: *She's the Man* is a 2006 American romantic comedy teen sports film directed by Andy Fickman and starring Amanda Bynes, Channing Tatum, Laura Ramsey, Vinnie Jones, and David Cross. (Fickman, 2006)

24.7 Advice to your former self

Okay good, thank you. So, any closing comments? Advice to your former self. So if you could speak to yourself in first year or to first or perhaps second year student, who is at the beginning of their studies, what would you advise them? I know you've been doing this in your roles as PASS leader. But what one piece of advice would you give them about getting the most out of being at university?

Carmen: Definitely getting involved? Because I feel like I took the most advantage of that I could and that was one thing. I know it's hard to not compare yourself to others. I know we constantly do it. And you're doing the labs and you do you may take, you see people who have been coding since they are tiny and they take two minutes to the labs. And we take, I don't have any experience, so I took a bit way longer than that. So, just understand that the fact that they take less time, doesn't mean that they will by the end of the degree. It doesn't mean that they'll land a better job because they've been coding for longer because it's also about the learning stuff.

Focusing on your also personal best, right? Yes. You know, not the Olympic gold medal standard? Yes. It's just saying, well, am I doing better than I did last time? Absolutely. And it's quite easy to fall in to that trap. I think especially at university where you're surrounded by other people who bright, motivated and hardworking. And if you compare themselves to those, it can lead to mental health problems (see chapter 3). I think, should we say? I'm not. Absolutely. It's very demotivating.

Carmen: So doing that and also balancing it out with the extra-curriculars it gives you also that extra, gives you more. Well, rounded for when you finish.

Well, thank you for joining us Carmen. Thank you and we'll see you at graduation.

Carmen: Yes, graduation. Yes.

See also *Running a Hackathon in lockdown*. (Práxedes, 2020)

24.8 Disclaimer

Coding Caution

Please note these transcripts are generated with speech to text software and are not perfect word-for-word transcriptions. Some speech disfluency has been manually removed and links, cross references and pictures have been manually added for clarification. Extra (non-quoted) words are given in brackets like this: (some text added afterwards).

Chapter 25

Sneha

Meet Sneha Kandane, shown in see figure 25.1 she graduated with a BSc degree in Computer Science with Industrial Experience in 2022.

Listen to the episode by clicking Play below, or subscribing wherever you get your podcasts, see section 20.1. An annotated and edited transcript of the audio is shown below.

25.1 What's your story Sneha?

So, Welcome Sneha, Thank you for coming on the show. Let's start with, tell us a little bit about yourself and where you come from?

Sneha: So my name is Sneha, like you said, I've just finished so I'm so excited. Just graduated my bachelor of Science in computer science with a year in industry from the University of Manchester. I'm actually from Manchester, this is my hometown and but originally grew up here.

You grew up in Manchester?

Sneha: since I'm twenty three years old, right? Well, okay. But I was originally born in India but like my whole life has been in Manchester. A little bit about me.

Good. Okay, so the next question is I'm really interested to find out why why people study computing because there's lots of routes into it as a subject. So what was it that made you think this is something I'd like to spend three or four years at university doing

Sneha: so it wasn't something I was always "oh this is my calling", you know, this is my this is the field of my choice. I was thinking of various things but I am it may not be as exciting of a story but



Figure 25.1: Sneha Kandane. Picture reused from [linkedin.com/in/sneha-kandane-931346183](https://www.linkedin.com/in/sneha-kandane-931346183) with permission, thanks Sneha.

genuinely just going through that that window. Sorry. Yeah, that's going past carry on. So, it's not as exciting, but that whole going through the education process from like GCSE and then to like a level and then, you know, at that point I I think I had that good firm like but you know that ground that footwork. So that's really helped because, you know, it made that I had that incremental kind of development when it came to the subject and then like obviously I enjoyed don't a bit of computing,

You did on a GCSE in computing?

Sneha: Yes. And A-level as well. Yeah, right. Okay, so I did both That's just because my school offered it I thought you know why not but before I didn't have any like in primary school and nothing but I think that really helped so because like I enjoyed it and then also like you know I did do well in it. I thought you know this is maybe like you know a good and it has obviously good prospect as a field and like another thing I always say is it has that right balance between the creativity and like you know problem solving and trying to get you to use your brain because I like my creative sides. I'm very like, you know, artistic I do a lot of artwork and things. And so this was like, you know, kind of an outlet in some sense as well. And like, that progressive nature of the subject means, like, you know, you start off small and you get bigger and bigger and bigger, and you get feedback as you go through those steps. So I really enjoyed, you know, right? Especially programming, you write a piece of code and then it does something and you kind of see it, you know, quite instantly like, you know, not always but like, quite most of the time usually, see it doing something and that gives you that gratification. So that was probably the reason why

Good. Okay. You studied computer science in your first year and then at the beginning of the second year you decided to do a placement. Was it something you decided before you came or is it something you decided when you got here? You thought actually I'd like to do a placement?

Sneha: So actually, so what happened was I was on the bachelor's, you know, your standard three year bachelor's course. And then I came into the interview day and that's when they introduced what an industrial placement is I never even knew what that meant and they explained what it was to me and I thought hey that actually sounds really useful. So on my interview day after that, I spoke to one of the people there and I said, like do you think I could change? So they told me to just email someone and that's how like I, you know, switched onto it.

25.2 Matillion

So the beginning of your second year, you started looking for placements or did you have any experience before then, any kind of in? So you going in? I think that's most people. I like that. Yeah.

Sneha: Kind of get your second year and it's like, I want to get some experience now.

So how did you go about choosing somewhere to work? The interesting thing is that different people have different obstacles they face when they find a job. So what were the obstacles you came across and how did you get around them?

So, like some obstacles, obviously, finding them . Finding one (where) they like you and you like them as well, obviously there's always that they have to choose you, but you also have to, choose them. So, it's that kind of liking towards the company. So there were companies, I applied for and things, but they, if I had got a placement, it would have just been for the sake of, getting a placement there rather than I actually enjoy, you know, I'd want to work there and I'm like, obviously you have your like challenges of like, you know, the interviews and the coding assessments (see section 10.2.3) and all of these like things especially for me because I never really did. This is like my first job like ever, like even like, you know, no part-time job previously. So, like this is all fresh new to me. So, just knowing how to navigate that. I made the mistake of not maybe applying to loads, I know loads people who just applied to like a ton. So if I'd go back, I would apply to way more than I did. So, you know, just and also, I'm very timid, so I got really scared and overwhelmed. So actually putting my foot forward to just apply. That was a challenge.

Jump in the water and see what happens?

Sneha: Exactly. That was quite challenging but and also the coding assessments were , obviously you do that throughout, you know, your years of programming, but actually being like, you know, this, you're gonna get a job based on this. It never experienced. That's, that was another challenge.

Yeah, definitely. So when what was I mean, I guess you, you must have had some unsuccessful interviews>

Sneha: Did you know, most people get quite a few rejections.

Don't they see how you had some rejections? I guess was location of factor. Use of trying to target a particular location. Or will you anywhere in the UK? Or I

I genuinely didn't mind. I mean, I I didn't apply outside of the UK. I was all within the UK but I wasn't like like to pick you if it was

like Manchester or elsewhere like I didn't mind. Right?

Okay.

Sneha: So the company ended up working for was a Matillion. It's in Manchester's.

So can you tell us about what Matillion does?

Sneha: So Matillion's focus is data transformation. (Extract, transform, load) So for cloud data warehouses, so it helps companies transform their data, especially, to move it to the cloud and it provides that kind of like middle kind of software (middleware) that like the product basically helps take the data, you know, extract it from wherever their company may have their data transform it like it allows the company to transform in a way they see fit. And then, you know, we work with like a AWS (Amazon Web Services) and Google BigQuery and my Microsoft Azure to like, you know, their day to warehouses, is to keep them all the data there. And I also other products help like analyse data and, like, in the cloud and help, you know, people be able to, like,

What was your particular role you there?

Sneha: I was a so going in as a placement software engineer. So, the good thing was, I wasn't even treated as a placement software engineer. I was just treated as a normal software engineer, in the sense that my work wasn't separate like in, like, oh, you can do this little project on the side. It was very much. You're on the product and you're gonna impact as much as anyone else. Yeah. That's good. To be to be a part of the team. Yeah, not

not sort of somebody's making cups of tea?

Sneha: Exactly.

So what would you say the main things you learn from your placement?

Sneha: You learn a whole lot of stuff at university, using the main things you've got out for your placement work. So like you for like so you have the educational side where that I learnt a lot about you know, software engineering practices because during uni you learn a lot of theory and learn a lot (about) developing your programming skills and everything. But you know in a practical industrial level, you never really see that side until maybe third year. But before that, I never really got that. So actually how the industry works, how a company a software company works that was what was introduced and I thought that was very interesting and, like different that definitely like, you know, it we did the whole software engineering module in year two. It was like kind of applying those principles but it also helped me kind of understand agile practices more which

really helped in third year and just how to co-operate with different team members and when you're working on such a big like, you know, this is not just a small repo, it's huge and you're working on a big product, like how that goes, what the process is like, obviously, you can't affect something. So, dangerously. So you, there's a whole protocol and you kind of learn about these things and, you know, what's the most efficient thing for a team. But also on the other side is just like personally I got a lot of communication like, you know, I built that communication a skill and then also just like ownership of the work that I did because I understood, you know, this is gonna impact something bigger than just, maybe like a coursework in like, you know, university. So you have to really put your effort into it and it's more than just something for you. It's something bigger. So, you know, taking responsibility and also the good thing about the company I work for is that they, you know, encouraged even me who's just like a placement, like, and software engineer. It's like, actually coming to meetings. Give my say and like that can actually have an impact on the company. So they, I think that's okay. And I guess a lot of cloud stuff as well because you've done some theory of the cloud stuff, but actually using it using AWS to in using GCP (Google Cloud Platform) and things like that to do to do stuff. Yes. Customers in your experience as well. Oh, for sure. So most enjoyable partner.

Maybe you've covered that already but what was the most enjoyable thing about working. Apart from being paid!

Sneha: That was amazing I think. So one thing that kind of was a bit sad for my placement, it was completely remote because of COVID Obviously. Yeah. So placement was just like all like I was just confined to my desk and the people yeah at home and people I met was just like my team and that was about it. But even despite that the social life of the company was like, they were really trying the hardest. So you know that actually did help because everyone is super friendly. I think also because it's not a huge company. Everyone's very homely, everyone just really wants to try and make everyone feel welcome. So I really enjoyed the culture that was the, I think that that's the biggest reason why I like this. Come, I chose the company because the culture was just super like sweet, everyone was nice and they really want to like better. You like you know, help you out. So, you everyone just willing to help no, major big pressure, Obviously, this responsibility, but nothing, you never felt bad about anything and that learning environment. They always encourage you to work. So once every month, they'd encourage us to just stop with our work and just learn something they we had udemy.com available to us. So I started learning courses on that. So I think it was just the extra things aside from just work. Yeah,

that really I made it

Culture is important I think some people sometimes lose sight of that. It's just kind of just look at the bottom line with how much are they paying?

Sneha Yeah.

And sometimes I think sometimes an interview can be quite hard to work out what the culture is actually like until you actually start there and but it was good to hear you had a good experience of that. So you came back to university, so come back from doing a placement back to university doing your final year project. And so can you tell us a little bit about your final year project? But for our listeners. Can you tell us a little bit about what you to feel?

Sneha: Yes, so my final year project was coding their future and my supervisor was you! So, the final year project basically looked into the current state of secondary school, education of computer science and the focus I wanted to take with it was identifying potential problems or issues with the current, state of the education potentially, even the curriculum and then finding actual practical solutions for teaching because currently, the uptake of it is very low. So how can we make that, you know, much better. And so the way that was done, was I worked with a school and a class and trialled different teaching methods, how different materials make create a different materials, used hardware and various other resources to enthuse the students to actually enjoy computer science rather than just have to do it because it's part of the school curriculum

Yes, did you find going back into school? Because it's very different, isn't it being at the front of the classroom? As opposed to being *in* the classroom as a student. So, how was that? How did you find that?

Sneha: it was actually surreal because I actually went back to my high school with my computer science teacher. So, it was really weird that now I'm working with my teacher rather than as a student. So, teaching the students, I think I was slightly like blessed in a sense, because the class were really nice. They were all willing to learn. They weren't super disruptive. Obviously, there were some characters, but, you know, the teacher was there to like, you know, calm down. Do the behaviour control kind of thing. But it was very interesting to see how they responded. And I was actually very surprised to see, like some of the trends, you know, of how they reacted to different things. And, but also because I taught year 9 class, some of them, that's when you choose your options, some of them kind of like, knew coming in, they weren't gonna pick computer science. So they didn't think to try. So that was slightly (a) struggle when you're trying to, you know, get them to do stuff but they don't want to because they know they aren't interested. So that was slightly, you know,

A bit of a battle.

sneha: Yeah.

Ongoing battle. I suppose?

sneha: Yeah

Good. Okay. So what comes next? So you're graduating next week?

sneha: Yes. Next in next Thursday

What are you planning to do after?

sneha: I'm obviously going to have some time off and relax a bit. I'm still going to work at Matillion but I won't be a placement software engineer. So I'll be a graduate software engineer. I'll be continuing to work with the same team on the same product but we're shifting to a different project so I'll be helping on the new project. It's a bit more of my knowledge on this so that's why they decided to keep me on it and also like I do plan on like you know while I'm at the company. I want to learn and they offer like you know, to go and do courses to keep it like agile courses or like AWS courses keep you know in that education field. Long-term wise I do you know, ideally I do want to do a masters in computer science. But I have to see like, you know what, the future holds. But I do want to get further, even further with education.

I think it's good to sometimes just have a little bit of a break, and I think it takes quite a long time to digest what you've done. It's so intense, especially during COVID as well. You've covered a lot of ground in the last four years and I think it just takes a while for some of those things to sink in and think, what am I doing next what am I interested in?

sneha: Exactly

It's no great, hurry to decide about going into postgraduate and I think sometimes having that experience of working will help you if you do decide to go and do postgraduate study

Sneha: that's what I was thinking. If you know, the feel like they value like experience a lot, I thought that would be better. And also you know you I've been studying from like, you know, proper like state school like exams from GCSE like year 10 all the way until now like it's a lot. So I definitely need that break before pursuing even further education.

Okay, so that's the degree bit your degree, your life in four years would cover that very quickly. There's a few other things to visit here. So they're one of them is minority Report. So you're a member of a minority group. You're a woman and you're also a British Asian. So I'd been interested in know what your experience have been has been when we've spoken to some people about

these things already. But the interesting know what your experiences have been as of that in university and then also in the workplace as well.

Sneha: So, with in terms of the university the good thing is, is like it's very diverse. So, for example, all the people that I've talked to especially my friendship group is super diverse in the sense backgrounds. So I have friends, literally from all the different continents you know, it's very diverse. So you see loads of different heritages and people and you learn a lot and we actually talk about these things and they're all different experiences. You're learning from each other which I think, in terms of diversity of like, culture and background. It's definitely like a good place for the university and then but like in terms of maybe of like, you know, as a woman I think I it there is like, kind of a struggle. Sometimes I'd say, in my year, there were quite a few.

I think, I need to look at the stats but I think it's something around about 25% of undergraduate students in computer science are women. So you're in a not a tiny minority, but still a minority. So I wonder if there's anything that they're not just this university but all universities can do to make the subject more welcoming or interesting or more desirable to women.

Sneha: That was part of my final project was to look at how to introduce, how to demystify, those stereotypes of the subject. you know, encourage females to take up the subject to enjoy the subject. Making it look show, maybe the more creative side, you know, catering to things that women tend to know like, yeah, I think there is that like that, you know, you can kind of tell, it's 25%, as I mean, if you like, look at like a lecture theatre or like, you'll be start, you know, like this big, yeah, obvious indication and even personally, even though there were likely in a lots of females I could talk to and stuff, most of my friends were male not because like I couldn't make friends with females but because there weren't like as many and, you know, sometimes as, you know, like a few more you want to see like especially, you know, your representation kind of you kind of want to feel a bit more comfortable knowing that you're not just the only one and also you're able to share experiences where because I think that's definitely a big thing you know, you as a woman and you as a male like they're they will be differences. And I feel like I could have had a bit more of a more comfortable feeling if I could have more female to relate to. But I know the, you know, especially in computer science, there are like initiatives to get that kind of going. So, you know, I can't say that there's nothing going on. I just think, you know, maybe a little bit more work in that sense. In terms of workplace, I can say that they are definitely trying

Everyone's trying. Well, *most* people are trying. Anyway. It's a hard problem, isn't it?

Sneha: it really is yeah, they try and introduce seminars or they had like a whole weekly meeting talking about, they got like, professional to talk about these things. But like, as much as you know, there is a lot of diversity. I feel like maybe in the engineering department, there's less diversity. Once again, predominantly male and actually, even predominantly, , white males and even more with recruitment to my team, the past three people that have come in, have all been white males and so that I know they are aware. They're recruitment is a little bit, you know, maybe not as diverse, and they're trying to change that. So I do I see that they are trying, but they're obviously things, you like kind of see and like, the reason is like, like the reason why I'd want to get a bit more diverse as a person like, you know, like from a minority group, it's just, you know, these kind of things they do impact your experience computer science or at work, at school. So you kind of want to be able to relate to other people in these experiences. So I think that's why like a super important to be able to see that. But what, in terms of teaching, a lot of the lecturers I do, they will a lot of female lecturers and even my like, supervisor from first and second was a female. So, like it would like, I think in terms of like staff and teaching that was quite good diversity.

25.3 One tune

So the last thing to finish up, well, the last bit to finish up is one one tune one podcast. And one film was the penultimately? Yeah. Is so, um, a piece of music.

There's lots of musical references in this textbook, code in your future. And do you have a piece of music that's important to you that you could add to our playlist? So when I was like reading this, it said it's mostly like, dad, rock. And I this funny thing is is that I'm not really gonna be going far from that.

Sneha: The song that I chose was a November Rain by Guns and Roses. (Rose, 1992)

Okay, good. Now that's good too good.

Sneha: It's very good song. Like the reason I think it was like, super important to me first of all like the, you know, the vibes of the song is very like impactful. Like you got you can listen to it when you're like sad or you can listen to when you like need to get pumped up like and the reason like it means so much to me is is that it's the song that for ages me my sister, my dad on the way to school, we would just listen to this and then on the way back we would listen to this. So it's very reminiscent and so that was the go-to song and then after a while I just like didn't like it's like you know, you kind of just forget about it and then what reminded me was the recent

Thor film and that was released thought.

Love and thunder and it's used on that it was (Waititi, 2022)

it was on the soundtrack and like this fight scene and it was just like, it made me feel like all the emotions like, you know, like just kind of gave me that nostalgia and I realised, oh my gosh, this is such a good song.

So as soon as after that, like me and my sister we put it back on our playlist because we like, you know, just realised like it's such a good song.

25.4 One podcast

Good choice. Okay. So the other thing and is one podcast. I don't know if you listen to podcast but other if there are good podcasts out there that you listen to would you is the one that you recommend?

I don't really tend to listen to many. But one podcast I do is rotten mango. (Soo, 2022) It's very, it's more. Like, if you're into true crime obviously dark topics, but she takes it from a psychological perspective and a legal perspective, all these different things, it's tied in a more lighthearted way.

Obviously in a more, it's still respectful. But in a very, you know, you're not feeling like groggy and like upset about the things you're listening to. It's very like interesting. So, if you're into that, it's really good. Okay. And then one film, was it gonna be the same film that you would just talk about using a different filter?

25.5 One film

I think if anyone asks me this is my film, it's called 3 idiots, see figure 25.2. (Hirani, 2009) It's a Bollywood film, but it's got a lot of global recognition.

So obviously it's in a different language. What sort of is, is it maybe made recently? Or is it? Oh no, it's almost like 2000. I mean not old all the 2009? Yeah, 10 years ago, he's old. So it's the best Like, obviously, because it's in a different language, you may think?

Oh no, like, can't really relate, but it's like generally, it just it crosses culture.

Does it have all the Bollywood dancing and the music?

Sneha It was it It is. The story is of like three guys who go through who get into like one of the best engineering schools in India and it deals with the pressures of like you know having to do well and it deals also with like different societal issues, you know, pressures of yourself, precious of your family, maybe like this is not the field.

3 Idiots is a 2009 Indian Hindi-language coming-of-age comedy-drama film written, edited and directed by Rajkumar Hirani

3 Idiots

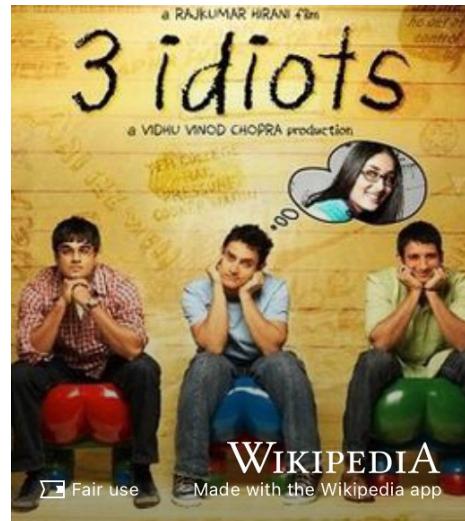


Figure 25.2: *3 Idiots* is a 2009 Indian Hindi-language coming-of-age comedy-drama film written, edited and directed by Rajkumar Hirani and co-written by Abhijat Joshi, with producer Vidhu Vinod Chopra. (Hirani, 2009)

You even want to go in or like all of these like very relatable topics that, you know, I feel like students do go through and but it and it deals with like a lot of like deep hard-hitting problems but it does all of that while still also being like the funniest film ever.

So it's like it's got the comedy, the drama. It's got the romance. It's got the suspense. Like, it's got everything. Like, that's why I recommend it strongly. I don't even speak Hindi. So like, I have to watch it with sometimes but still I, I don't watch films like over and over again.

But I watched this one like five for six times. Crazy actually.

25.6 Advice to your former self

I'll make a note of that one definitely. So the last thing to finish up then is I call this time travel. So, if you could travel back in time to meet yourself in the first year, what advice would you give yourself or perhaps current or incoming first, second year, students. What advice would you give them to get the most out of their time at university?

Sneha: So I think to two things. So the first one is just putting yourself out there and taking risks because more often than not, it's rewarding. Because even if like it's not your even if you don't get your like intended result, the fact that you've done it and you've

tried is better than not knowing if you didn't try so like especially like when I was applying for like placements and stuff completely out of my like, you know what?

Sneha: I know. So I when I did that, you know, obviously after some time, like I did get that reward from it. So just really putting yourself out of your comfort zone. Going to things and I'd say, because obviously with COVID and stuff. Like, you know, it was a bit more difficult but even like socially just like, you know, trying to put yourself out there as an introvert. That's very scary to me. But it's more often than not there's pay off from it. So that's the first piece and then the second one is just hanging in there. You know persevering because to think that my university journey was super smooth sailing is very naive. I remember first year, obviously University of Manchester has really high standards, especially in the computer science department, and that means with such high standards. You kind of doubt yourself, you think maybe this isn't for me and that was definitely something I went through. First year, I was even looking at maybe I should apply for a different course. I think the thing that consolidated, yes, this is your field. This is what, you know, you want to do, was my placement. Yeah, actually,

Right

Sneha: After my placement, I knew I actually enjoyed this. I can do for the rest of my life like this. That consolidated it for me. And so even with struggles and coursework and pressure and eventually those things will finish and those things, you know, you will overcome and then not more often than not. They're not gonna impact your life. You know. Long-term so like is you just kind of have to push through that bit? Yeah, now I'm sitting here about to graduate, so it does pay off just pushing through

That's good advice. Okay, good. Well, thank you, thank you for joining us today

Sneha so much for having me

we look forward to seeing you at graduation next week.

Sneha: Yeah, so exciting. Okay. Thank you.

25.7 Disclaimer

Coding Caution

Please note these transcripts are generated with speech to text software and are not perfect word-for-word transcriptions. Some speech disfluency has been manually removed and links, cross references and pictures have been manually

added for clarification. Extra (non-quoted) words are given in brackets like this: (some text added afterwards).

Chapter 26

Alice

Meet Alice Păcuraru, see figure 26.1 she graduated with a Bachelor of Science degree in Computer Science with Business and Management and Industrial Experience in 2022.

26.1 What's your story Alice?

Podcast episode and transcript to appear here



Figure 26.1: Alice Păcuraru. Picture reused from [linkedin.com/in/alexandra-pacuraru](https://www.linkedin.com/in/alexandra-pacuraru) with permission, thanks Alice.

Chapter 27

Reading your future

If you want to have your future read, you can read all about it right here. There is no need to gaze into any crystal balls! These books, journals, websites, magazines will help you to read your future. Reading is good for your mind, body and soul too because libraries give us power (Bradfield et al., 1996; Moore, 2019), see figure 27.1.



Figure 27.1: Libraries give you power, the power to read your future. These references are a digital library you can search and browse. Panorama of the British Museum Reading Room by David Iliff on Wikimedia Commons [w.wiki/3BE](https://commons.wikimedia.org/w/index.php?curid=3BE)

This chapter lists everything (and I mean *everything*) cited in this book which you might find overwhelming. For a more easily digestible versions see the shorter “signposts” section of any chapter in this book and see Gavin Brown’s chapter on how to read critically. Critical thinking and critical reading are valuable skills, not just for PhD students and not just for reading scientific papers either. So as Bill Gates puts it “read a lot” (see figure 27.2) but read *critically* too. (Brown, 2021b)

Since you are reading the pdf version, all the references can be found in the

My advice is simple:
Read a lot and discover a
skill you enjoy

Bill Gates



Figure 27.2: “I was recently asked what advice I would give to young people who want to make a positive impact on the world ... My advice is simple: Read a lot and discover a skill you enjoy. For some, that means being great at science or a great communicator. There’s so much opportunity to do good in the world.” (Gates, 2022) CC BY portrait of Bill Gates by Kuhlmann / MSC on Wikimedia Commons w.wiki/3W7k

bibliography section following this page.

Bibliography

- Aaron, S. (2016). Programming as performance. <https://youtu.be/0lTZ8Tuyu5I>.
- Agnew, H. (2016). Big four look beyond academics: Firms want to make offers based on potential, rather than personal circumstance. <https://on.ft.com/3bIbVBz>.
- Aicher, T. P., Barabási, D. L., Harris, B. D., Nadig, A., and Williams, K. L. (2017). Ten simple rules for getting the most out of a summer laboratory internship. *PLOS Computational Biology*, 13(8):e1005606.
- Al-Khalili, J. (2022). Frances arnold: From taxi driver to nobel prize. <https://bbc.in/3TXFXSP>.
- Anderson, C. (2016). *TED Talks: The Official TED Guide to Public Speaking*. Nicholas Breley.
- Andreessen, M. (2011). Why software is eating the world. The Wall Street Journal <https://a16z.com/2011/08/20/why-software-is-eating-the-world/>.
- Anon (1611). The gospel according to st. matthew. In Carroll, R. and Prickett, S., editors, *The Bible: Authorized King James Version*. Oxford University Press.
- Anon (1744). Baa, baa, black sheep. In Anon, editor, *Tommy Thumb's Pretty Song Book*. Ah! vous dirai-je, maman, Mary Cooper.
- Anon (1745). God save the queen. In Anon, editor, *The National Anthem of the United Kingdom*. Monarchy in the UK, The Gentleman's Magazine.
- Anon (2010). Cheltenham ladies' college principal vicky tuck resigns. <https://bbc.in/3A7QJ06>.
- Anon (2014). Reflective writing. <https://youtu.be/QoI67VeE3ds>.
- Anon (2019a). Generation Z is stressed, depressed and exam-obsessed: For most youngsters getting good grades is a bigger worry than drinking or unplanned pregnancies. <https://econ.st/3ArLVns>.

- Anon (2019b). Open Badges v2.0: IMS Final Release. <http://www.imsglobal.org/sites/default/files/Badges/OBV2p0Final>.
- Anon (2021). Computing for net zero: how digital technology can create a ‘control loop for the protection of the planet’. <https://royalsociety.org/-/media/policy/projects/climate-change-science-solutions/climate-science-solutions-computing.pdf>.
- Anon (2022). Editorial: How AlphaFold can realize AI’s full potential in structural biology. *Nature*, 608(7921):8–8.
- Apostoleanu, E. A. (2018). Iguana. In Ciente, D., editor, *Yo. INNA*, Mushroom Records.
- Appleby, S. (2008). How to Draw with Steven Appleby. https://youtu.be/G_Vz7gTh1VE.
- Ashcroft, R. (1997). The drugs don’t work. In Potter, C., editor, *Urban Hymns*. The Verve, Hut Records.
- Assay, M. (2020). Want an IT job? look outside the tech industry. <https://www.techrepublic.com/article/want-an-it-job-look-outside-the-tech-industry/>.
- Association, N. (2016). How did calling nightline help me? <https://youtu.be/RsfNkSZmUc8>.
- Babineaux, R. (2013). *Fail Fast, Fail Often: How Losing Can Help You Win*. Tarcher.
- Bacon, K. (1994). Six degrees of kevin bacon. https://en.wikipedia.org/wiki/Six_Degrees_of_Kevin_Bacon.
- Ball, C. (2022a). Busting graduate job myths. https://www.universitiesuk.ac.uk/sites/default/files/field/downloads/2022-04/busting-graduate-job-myths_0.pdf.
- Ball, C. (2022b). UK graduate labour market update. <https://luminate.prospects.ac.uk/uk-graduate-labour-market-update-11-january>.
- Barrass, R. (2002). *Scientists Must Write: A Guide to Better Writing for Scientists, Engineers and Students*. Routledge.
- Barshay, J. (2022). College students often don’t know when they’re learning: Harvard experiment reveals the psychological grip of lectures. <https://hechingerreport.org/proof-points-college-students-often-dont-know-when-theyre-learning/>.
- Bartlett, J. (2018). *The People Vs Tech: How the internet is killing democracy (and how we can save it)*. Ebury Press.
- Bates, L. (2016). *Everyday Sexism: The Project that Inspired a Worldwide Movement*. A Thomas Dunne Book for St. Martin’s Griffin.

- Bavaro, J. and McDowell, G. L. (2021). *Cracking the PM Career: The Skills, Frameworks, and Practices to Become a Great Product Manager*. CareerCup.
- Beaubouef, T. (2003). Why computer science students need language. *ACM SIGCSE Bulletin*, 35(4):51–54.
- Bell, C. (2021). Am I selling my soul to work for my company? <https://www.thevectorimpact.com/selling-your-soul/>.
- Belshaw, D. (2019). 10 platforms for issuing open badges. <https://blog.weareopen.coop/10-platforms-for-issuing-open-badges-f249cff609d42>.
- Berry, K. (1998). *Bradford on Avon's Schools: The Story of Education in a Small Wiltshire Town*. Ex Libris Press.
- Birchall, M. (2022). The graduate market in 2022: Annual review of graduate vacancies & starting salaries at the UK's leading employers. https://www.highfliers.co.uk/download/2022/graduate_market/GM22-report.pdf.
- Black, C. (1985). Blind date: What's your name and where d'ya come from? [https://en.wikipedia.org/wiki/Blind_Date_\(British_game_show\)](https://en.wikipedia.org/wiki/Blind_Date_(British_game_show)).
- Black, J. (2019a). *How to Find the Career You've Always Wanted: How to take control of your career plan – and make it happen*. Robinson. Previous title: Where am I Going and Can I Have a Map? How to take control of your career plan - and make it happen.
- Black, J. (2019b). How to write a top-notch CV. <https://youtu.be/yjdvCHWVtE4>.
- Blair, J. H. (1958). Blue peter. https://en.wikipedia.org/wiki/Blue_Peter.
- Blowers, A. (2015). Uptown funk. <https://youtu.be/z8qH05teRMM>.
- Bolles, R. N. (2014). Deciding what to do five years from now. <https://youtu.be/oeP6Pm3Xf-8>.
- Bolles, R. N. (2019). *What Color Is Your Parachute? A Practical Manual for job-hunters and career-changers*. Random House USA Inc.
- Borrett, A. (2019). Is a first class degree really that important? many graduate recruiters are weighing up other skills alongside academic achievement. <https://on.ft.com/3Cd4IEu>.
- Bourne, P. E. (2005). Ten simple rules for getting published. *PLOS Computational Biology*, 1(5):e57.
- Bourne, P. E., Lewitter, F., Markel, S., and Papin, J. A. (2018). One thousand simple rules. *PLOS Computational Biology*, 14(12):e1006670.
- Box, H. and Mocene-McQueen, J. (2019). *How Your Story Sets You Free*. Chronicle Books.

- Brackenbury, L. (2005). Digital 60: An interview with Linda Brackenbury. <https://bbc.in/2Eqyo1K>.
- Bradfield, J. D., Moore, S., and Wire, N. (1996). A design for life. In Eringa, D. and Hedges, M., editors, *Everything Must Go*. Manic Street Preachers, Def Jam.
- Bradshaw, T., Venkataramakrishnan, S., Moise, I., Franklin, J., and Silverman, G. (2022). Apple sidelines Goldman Sachs and goes in-house for lending service. <https://on.ft.com/3Qzomic>.
- Brand, R. (2022). Under the skin. https://en.wikipedia.org/wiki/Under_the_Skin_with_Russell_Brand.
- Brewton, S. (2014). Squire Bill Widener vs. Theodore Roosevelt. <https://suebrewton.com/2014/12/31/squire-bill-widener-vs-theodore-roosevelt/>.
- Britton, B. (2019). No sexuality please, we're scientists. <https://youtu.be/NEqfP1diyes>.
- Brown, G. (2021a). *How to get your PhD: A Handbook for the Journey*. Oxford University Press.
- Brown, G. (2021b). How to get your phd: A handbook for the journey. <https://profgavinbrown.github.io/booksamples/Chapter7howtoread.pdf>.
- Brown, I. and Squire, J. (1989). I wanna be adored. In Leckie, J., editor, *The Stone Roses*. The Stone Roses, Silvertone Records.
- Brown, N. and Wilson, G. (2018). Ten quick tips for teaching programming. *PLOS Computational Biology*, 14(4):e1006023.
- Burkeman, O. (2014). This column will change your life: Morning pages. <https://www.theguardian.com/lifeandstyle/2014/oct/03/morning-pages-change-your-life-oliver-burkeman>.
- Burkeman, O. (2021). *Four Thousand Weeks: Time Management for Mortals*. Vintage.
- Cailliau, R. (2014). WWW and more. In *From Physics to Daily Life*, pages 61–80. Wiley-VCH Verlag GmbH & Co. KGaA.
- Cameron, J. (1992). *The Artist's Way: A Spiritual Path To Higher Creativity*. Penguin.
- Carroll, D. (2018). What is chippy tea? <https://www.quora.com/What-is-chippy-tea>.
- Carroll, L. (1865). *Alice's Adventures in Wonderland*. Macmillan.
- Carson, S. (2022). Loud. In *Sofia Carson*. Sofia Carson, Hollywood Records.
- Carter, J. (2021). *Work placements, internships and applied social research*. SAGE.

- Charette, R. N. (2005). Why software fails. *IEEE Spectrum*, 42(9):42–49.
- Cheary, M. (2021). Hobbies and interests: Should I include them in my CV? <https://www.reed.co.uk/career-advice/hobbies-and-interests-should-i-include-them-in-my-cv>.
- Chen, C. (2020). Online learning is growing faster than ever — we compared 4 of the top platforms to help you decide which one makes sense for you. <https://www.businessinsider.com/online-learning-platform-comparison-udemy-skillshare-lynda-coursera>.
- Cliff, D. (2018a). Professor dave cliff: My mental health. <https://engineering.blogs.bristol.ac.uk/my-mental-health-an-honest-chat-with-professor-dave-cliff/>.
- Cliff, D. (2018b). This challenge of mental health. <https://youtu.be/OtELvV8-OK0>.
- Clinton, H. (1996). *It Takes a Village: And Other Lessons Children Teach Us*. Simon & Schuster.
- Cockburn, M. (2022). Ceo secrets: 'make decisions as if you were on your deathbed'. <https://bbc.in/3dDivtt>.
- Code, N. (2022). Change your life - algorithms and data structures for beginners. <https://youtu.be/TKC98Sumg2A>.
- Cohen, L. (1984). Hallelujah. In Lissauer, J., editor, *Various Positions*. Leonard Cohen, Columbia Records.
- Copeland, J. and Long, J. (2016). Restoring the first recording of computer music. <https://blogs.bl.uk/sound-and-vision/2016/09/restoring-the-first-recording-of-computer-music.html>.
- Copeland, J. and Long, J. (2017). *Turing and the History of Computer Music*, pages 189–218. Springer International Publishing.
- Coughlan, S. (2007). 'education, education, education'. <http://news.bbc.co.uk/1/hi/education/6564933.stm>.
- Coughlan, S. (2019a). 'grade inflation' means 80% more top degree grades. <https://bbc.in/3w914rg>.
- Coughlan, S. (2019b). The symbolic target of 50% at university reached. <https://bbc.in/3AqCrJ0>.
- Coyne, W., Drozd, S., Ivins, M., and Fridmann, D. (2002). Do you realize?? In Booker, S., editor, *Yoshimi Battles the Pink Robots*. The Flaming Lips, Warner Records.
- Curtis, I., Sumner, B., Hook, P., and Morris, S. (1980). Isolation. In Hannett, M., editor, *Closer*. Joy Division, Factory Records.

- Cutts, Q., Cutts, E., Draper, S., O'Donnell, P., and Saffrey, P. (2010). Manipulating mindset to positively influence introductory programming performance. In *Proceedings of the 41st ACM technical symposium on Computer science education - SIGCSE '10*. Association for Computing Machinery.
- Dahl, R. (1964). Charlie and the chocolate factory.
- Dale, S. (2018). #NotJustLondon — what's that then? <https://medium.com/21st-century-mindset/notjustlondon-whats-that-then-a2c5822eadc9>.
- Damore, J. (2017). Google's ideological echo chamber. https://en.wikipedia.org/wiki/Google%27s_Ideological_Echo_Chamber.
- Danton, T. (2021). *The Computers That Made Britain: Inside the home computer revolution of the 1980s*. Raspberry Pi Press.
- David, J. (2017). How long should your CV be? <https://www.cv-library.co.uk/career-advice/cv/how-long-cv-be/>.
- Davidova, P. (2021). My worst software engineering interview fails: Failing my facebook and google interviews. <https://youtu.be/qkeQNNjZuQk>.
- Davis, V. W. (2016). Error reflection: Embracing growth mindset in the general music classroom. *General Music Today*, 30(2):11–17.
- Desai, R. (2016). Generalized anxiety disorder (gad) - causes, symptoms and treatment. <https://youtu.be/9mPwQTiMSj8>.
- Deslauriers, L., McCarty, L. S., Miller, K., Callaghan, K., and Kestin, G. (2019). Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proceedings of the National Academy of Sciences*, 116(39):19251–19257.
- Dick, P. K. (1956). *The Minority Report*. Fantastic Universe.
- Downes, G., Horn, T., and Woolley, B. (1979). Video killed the radio star. In Hurst, M., editor, *The Age of Plastic*. The Buggles, Island Records.
- Dweck, C. (2014). The power of not yet. <https://youtu.be/J-swZaKN2Ic>.
- Dweck, C. (2017). *Mindset: Changing The Way You think To Fulfil Your Potential*. Robinson.
- Dylan, B. and Hewson, P. (1988). All along the watchtower. In Iovine, J., editor, *Rattle and Hum*. U2, Island Records. album.
- Easton, F. (1997). Educating the whole child, “head, heart, and hands”: Learning from the waldorf experience. *Theory Into Practice*, 36(2):87–94.
- Eddo-Lodge, R. (2017). *Why I'm No Longer Talking to White People About Race*. Bloomsbury Publishing.
- Elgar, E. (1912). The music makers. In Richter, H., editor, *Birmingham Triennial Music Festival*. London Symphony Orchestra, EMI Records.

- Evans, J. (2018). Help! I have a manger! how to understand your manager's job and do awesome work together. <https://wizardzines.com/zines/manager/>.
- Evans, J. (2021a). Hell Yes! CSS! <https://wizardzines.com/zines/css/>.
- Evans, J. (2021b). How containers work. <https://wizardzines.com/zines/containers/>.
- Evans, J. (2021c). How DNS works. <https://wizardzines.com/zines/dns/>.
- Evans, J. (2021d). HTTP: Learn your browser's language. <https://wizardzines.com/zines/http/>.
- Fauvel, J., Flood, R., and Wilson, R., editors (2006). *Music and Mathematics: From Pythagoras to Fractals*. Oxford University Press.
- Favreau, J. (2008). Iron man. [https://en.wikipedia.org/wiki/Iron_Man_\(2008_film\)](https://en.wikipedia.org/wiki/Iron_Man_(2008_film)).
- Ferrari, V., Marin-Jimenez, M., and Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.
- Feynman, R. (1988). Feynman's blackboard at time of his death. <https://calisphere.org/item/b3e8d3cb9b8adc01314dba1b1f1fcf84/>.
- Fickman, A. (2006). She's the man. https://en.wikipedia.org/wiki/She's_the_Man.
- Fincher, S. and Finlay, J. (2016). *Computing Graduate Employability: Sharing Practice*. Council of Professors and Heads of Computing CPHC.
- Fitzpatrick, B. and Collins-Sussman, B. (2009). Google i/o: The myth of the genius programmer. <https://youtu.be/0SARbwvhupQ>.
- Flowers, B., Keuning, D., Stoermer, M., and Jr., R. V. (2004). Glamorous indie rock & roll. In Saltzman, J., editor, *Hot Fuss*. The Killers, Lizard King Records.
- Fogarty, T. (2015). Hackathons are for beginners. <https://medium.com/tfogo/hackathons-are-for-beginners-77a9c9c0e000>.
- Fonarow, W. (2010). Ask the indie professor: Why the 'best' indie is depressing, local and unpopular. <https://www.theguardian.com/music/musicblog/2010/sep/14/indie-professor-this-town>.
- Friedman, S. and Laurison, D. (2020). *The Class Ceiling: Why it Pays to be Privileged*. Policy Press.
- Fry, S. (2010). *The Fry Chronicles: An autobiography*. Penguin.
- Fry, S. (2014). We are mind. <https://youtu.be/OPxHTZRwXX8>.
- Fry, S. (2017). *Mythos: The Greek Myths Retold*. Michael Joseph.

- Fry, S. (2018). *Heroes: The myths of the Ancient Greek heroes retold*. Penguin Random House UK.
- Furber, S. (2000). *ARM system-on-chip architecture*. Pearson Education, 2 edition.
- Gabriel, P. and Bush, K. (1986). Don't give up. In Lanois, D., editor, *So. Peter Gabriel*, Geffen.
- Gallagher, N. (1995). Morning glory. In McGee, A., editor, (*What's the Story Morning Glory?*) Oasis, Creation Records Ltd.
- Garner, R. (2015). Ey: Firm says it will not longer consider degrees or a-level results when assessing employees. <https://www.independent.co.uk/news/education/education-news/ey-firm-says-it-will-not-longer-consider-degrees-or-level-results-when-assessing-employees-10436355.html>.
- Garone, E. (2014). To print or not to print — a CV, that is. <https://bbc.in/3zXv6zr>.
- Gates, B. (2022). What advice would I give to young people who want to make a positive impact on the world? <https://www.linkedin.com/feed/update/urn:li:activity:6943628912300101632/>.
- Gill, E. and Statham, N. (2021). Head who promised stability for 'forgotten school' with nine leaders in 10 years is leaving. <https://www.manchestereveningnews.co.uk/news/greater-manchester-news/head-who-promised-stability-forgotten-19703357>.
- Goble, C. (2007). The seven deadly sins of bioinformatics. In *Bioinformatics Open Source Conference (BOSC) Special Interest Group (SIG) at the 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB)*, Vienna, Austria. Slideshare.
- Goble, C. (2020). Celebrating leading women in engineering: Professor carole goble cbe. <https://web.archive.org/web/20200615015706/https://raeng.org.uk/diversity-in-engineering/diversity-and-inclusion-at-the-academy/celebrating-leading-women-in-engineering/carole-globe>.
- Goble, C. and Wroe, C. (2004). The montagues and the capulets. *Comparative and Functional Genomics*, 5(8):623–632.
- Goldman, P. and Schlesinger, Y. (2018). With great code comes great responsibility: Announcing the responsible computer science challenge. <https://medium.com/omidyar-network/with-great-code-comes-great-responsibility-announcing-the-responsible-computer-science-challenge-93cbee94b923>.
- Googler, A. (2022). Google summer of code student guide. <https://github.io/gsocguides/student/>.

- Gove, M. (2014). We must break down 'berlin wall' between state and private schools. <https://www.theguardian.com/education/2014/feb/03/michael-gove-berlin-wall-state-private-schools>.
- Graham, P. (2005). Good and bad procrastination. <http://www.paulgraham.com/procrastination.html>.
- Granovetter, M. S. (1973). The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380.
- Green, F. and Kynaston, D. (2019). *Engines of Privilege: Britain's Private School Problem*. Bloomsbury Publishing.
- Groom, B. (2022). *Northerners: A History, from the Ice Age to the Present Day*. Harper North.
- Hacker, A. (2021). International Collegiate Programming Contest (ICPC). <https://icpc.global>.
- Haig, M. (2016). *Reasons to Stay Alive*. Canongate Books Ltd.
- Haig, M. (2018). Top 5 tips for good mental health in a social media age. <https://youtu.be/JMm5kiiMjTM>.
- Haig, M. (2019). *Notes On A Nervous Planet*. Canongate Books Ltd.
- Haim, A., Haim, D., and Haim, E. (2012). The wire. In Rechtshaid, A., editor, *Days Are Gone*. Haim, Polydor Records.
- Hamedy, R. (2019). How to build a reputable StackOverflow profile. <https://medium.com/@rhamedy/contribution-debt-why-how-to-contribute-to-stack-overflow-a69d4bd50d0c>.
- Hampton, M., Hopes, L., and Hampton, L. (2020). Still moving: No new worlds. <https://stillmoving.org/projects/speedwell-no-new-worlds/>.
- Hankin, C. (2022). Smart cities. *ACM TechBrief*.
- Hardy, T. (1874). *Far from the Madding Crowd*. Cornhill Magazine.
- Harkleroad, L. (2006). *The Math Behind the Music*. Cambridge University Press.
- Harrison, G. (1968). While my guitar gently weeps. In Martin, G., editor, *The White Album*. The Beatles, Apple Records.
- Healy, F. (1999). Why does it always rain on me? In Hedges, M., editor, *The Man Who*. Travis, Independiente.
- Hendrix, J. (1967). Are you experienced? In Chandler, C., editor, *Are You Experienced?* The Jimi Hendrix Experience, Track Records.
- Hinds, D. (2017). Jobseekers being targeted by scammers. <https://bbc.in/3ArRbHV>.

- Hirani, R. (2009). 3 idiots. https://en.wikipedia.org/wiki/3_Idiots.
- Hitchens, C. (1997). Everyone has a book inside them. <https://youtu.be/PowW35ZryYA>.
- Hoffman, R. (2020). Customize your public profile URL. <https://www.linkedin.com/help/linkedin/topics/6042/6054/87>.
- Hopes, L., Hampton, M., and Hampton, L. (2020). Re-informed: Still/moving. <https://www.youtube.com/watch?v=Y2Ya4d1oSQc>.
- Horton, A. P. (2020). Could micro-credentials compete with traditional degrees? <https://bbc.in/3T5Ejyq>.
- Howard, H. (1951). Country music hall of fame and museum: Three chords and the truth. <https://web.archive.org/web/20200926193611/https://www.countrymusichalloffame.org/content/uploads/2019/05/WM-3-6-Harlan-Howard-Bio.pdf>.
- Hull, D. (2007). Scifoo day 1: Turn up, tune in, drop out. <https://duncan.hull.name/2007/08/04/scifoo-day-1-turn-up-tune-in-drop-out/>.
- Hull, D. (2009). California googlin'. <https://duncan.hull.name/2009/07/08/california-googlin/>.
- Hull, D. (2019). Are liverpool and manchester still in lancashire?
- Hull, D. (2020). Seven things to do at CERN if you're not a physicist. <https://duncan.hull.name/2020/01/27/cern/>.
- Hull, D. (2021a). How long should your cv be: One page resume or two page CV? <https://youtu.be/0abDOKHS5T0>.
- Hull, D. (2021b). Thank you NHS. <https://duncan.hull.name/2021/03/22/thank-you-nhs/>.
- Hull, D. (2022). Getting started with latex: A laboratory manual. <https://latex4year1.netlify.app>.
- Hull, D., Zolin, E., Bovykin, A., Horrocks, I., Sattler, U., and Stevens, R. (2006). Deciding semantic matching of stateless services. *AAAI 2006*, pages 1319–1324.
- Hütter, R. and Hilpert, F. (2003). Elektro kardiogramm. In Schneider, F., editor, *Tour de France Soundtracks*. Kraftwerk, Capitol Records.
- Investor, A. (2019). Let's make work better. <https://www.investorsinpeople.com/wp-content/uploads/2019/10/We-invest-in-people-framework.pdf>.
- Investor, A. (2020). How to bring yourself to work and allow others to do the same. <https://www.investorsinpeople.com/knowledge/how-to-bring-yourself-to-work-and-allow-others-to-do-the-same/>.

- Isley, O., Isley, R., and Isley, R. (1988). Shake your thang. In *A Salt with a Deadly Pepa*. Salt-N-Pepa, London Records.
- James, R. D. (1992). We are the music makers. In James, R. D., editor, *Selected Ambient Works 85–92*. Aphex Twin, Apollo Records.
- Johnson, H. (2020). The 'forgotten school' with nine head teachers in 10 years - and the man with a plan to turn it around. <https://www.manchestereveningnews.co.uk/news/greater-manchester-news/forgotten-school-nine-head-teachers-17830691>.
- Johnson, R. (2004). X.S.L.T. to the tune of Y.M.C.A. <http://lists.xml.org/archives/xml-dev/200408/msg00019.html>.
- Johnstone, M. (2012). I had a black dog, his name was depression. <https://youtu.be/XiCrniLQGYc>.
- Jr., J. D. M. (1988). Freak scene. In Kolderie, P. Q. and Slade, S., editors, *Bug*. Dinosaur Jr., SST Records.
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.
- Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus & Giroux, New York, NY.
- Kelion, L. (2018). Google's job hunting service comes to UK. <https://bbc.in/3QAOSba>.
- Knowles, B., Widdicks, K., Blair, G., Berners-Lee, M., and Friday, A. (2022). Our house is on fire: The climate emergency and computing's responsibility. *Communications of the ACM*, 65(6):38–40.
- Knuth, D. E. (1984). Literate programming. *The Computer Journal*, 27(2):97–111.
- Kohn, A. (2018). *Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes*. Houghton Mifflin.
- Kottke, J. (2018). Alan Turing was an excellent runner. <https://kottke.org/18/04/alan-turing-was-an-excellent-runner>.
- Lahey, J. (2016). *The Gift of Failure: How the Best Parents Learn to Let Go So Their Children Can Succeed*. Harper.
- Lane, R. and Wood, R. (1973). Ooh la la. In *Ooh La La*. The Faces, Warner Bros.

- Lanier, J. (2018). *Ten Arguments For Deleting Your Social Media Accounts Right Now*. Bodley Head.
- Lavington, S. (1998). *A history of Manchester computers*. British Computer Society, 2nd edition.
- Leendertz, L. (2006). *The half-hour allotment: Royal Horticultural Society*. Frances Lincoln, London.
- Lennon, J. and McCartney, P. (1967). A day in the life. In Martin, G., editor, *Sgt. Pepper's Lonely Hearts Club Band*. The Beatles, Parlophone.
- Leslie, I. (2020). Why your ‘weak-tie’ friendships may mean more than you think. <https://bbc.in/3QMme6j>.
- Lewis, P. (2017). ‘i see things differently’: James damore on his autism and the google memo. <https://www.theguardian.com/technology/2017/nov/16/james-damore-google-memo-interview-autism-regrets>.
- Lindsay-Hogg, M. (1996). The rolling stones rock and roll circus. https://en.wikipedia.org/wiki/The_Rolling_Stones_Rock_and_Roll_Circus.
- Link, D. (2012). Programming enter: Christopher strachey’s draughts program. In *RESURRECTION: The Bulletin of the Computer Conservation Society*, number 60. Computer Conservation Society.
- Longfella (2017). This is the place. <https://forevermanchester.com/this-is-the-place-fm/>.
- Louise (2019). Why I regret doing an unpaid internship. <https://www.ratemyplacement.co.uk/blog/why-i-regret-doing-an-unpaid-internship/>.
- Louise (2022). The highest paid internships and placements in the uk. <https://www.ratemyplacement.co.uk/blog/highest-paid-internships-placements-uk/>.
- Lydon, J. and Laswell, B. (1986). Rise. In Lydon, J. and Laswell, B., editors, *Album*. Public Image Ltd, Virgin Records.
- Malan, D. J. (2010). Reinventing CS50. In *Proceedings of the 41st ACM technical symposium on Computer science education - SIGCSE '10*. Association for Computing Machinery.
- Malan, D. J. (2017). CS50 at scale: University of Manchester. <https://youtu.be/CPWSTMhcFBA>.
- Malan, D. J. (2018). This is CS50. <https://youtu.be/3oFAJtFE8YU>.
- Malan, D. J. (2020). Teaching from home via zoom. <https://medium.com/@cs50/teaching-from-home-via-zoom-c3b336446fbc>.
- Malan, D. J. (2021). Toward an ungraded CS50. In *Proceedings of the 52nd ACM technical symposium on Computer science education - SIGCSE '21*. Association for Computing Machinery.

- Malham, G. (2002). Soul sold for less than £12. <https://bbc.in/3w8ML6d>.
- Mann, C. C. (2002). Why software is so bad: For years we've tolerated buggy, bloated, badly organized computer programs. but soon, we'll innovate, litigate and regulate them into reliability. *MIT Technology Review*.
- Manone, W. (1935). In the mood. In Miller, G., editor, *Tar Paper Stomp*. Big Band, Champion Records.
- Manson, S., Erikson, D., Marker, S., and Vig, B. (1995). Only happy when it rains. In Garbage, editor, *Garbage*. Garbage, Mushroom Records.
- Marchese, D. (2022). Yale's happiness professor says anxiety is destroying her students. <https://nyti.ms/3po3MFv>.
- Markow, W., Coutinho, J., and Bundy, A. (2019). Beyond tech: The rising demand for IT skills in non-tech industries. <https://www.burning-glass.com/research-project/beyond-tech/>.
- Marr, J. and Morrissey (1984). Heaven knows i'm miserable now. In Porter, J., editor, *Hatful of Hollow*. The Smiths, Rough Trade Records.
- Marsden, G. (1964). Ferry cross the mersey. In Martin, G., editor, *Ferry Cross the Mersey*. Gerry and the Pacemakers, Columbia Graphophone Company.
- Martin, R. C. (2011). *The Clean Coder: A Code of Conduct for Professional Programmers*. Prentice Hall.
- Massey, G., Barker, A., Price, M., and Partington, D. (1991). Nephatitis. In Haas, M., editor, *ex:eI*. 808 State, ZTT Records.
- Mathers, B. (2021). Tedxgalway: People tell stories; stories paint pictures. <https://youtu.be/IapGM5ZYBEw>.
- Maya and Emma (2021). Me time. <https://spoti.fi/3QKAAEo>.
- McCormack, D. (2017). Some lesser-known truths about academe. *The Chronicle of Higher Education*.
- McDowell, G. L. (2011). Ex-googler reveals strategies to land interview with Google. <https://www.cnbc.com/id/42608205>.
- McDowell, G. L. (2014). *Cracking the Tech Career*. John Wiley & Sons Inc.
- McDowell, G. L. (2015). *Cracking the Coding Interview: 189 Programming Questions and Solutions*. CareerCup, 6 edition.
- McDowell, G. L. and Bavaro, J. (2013). *Cracking the PM Interview: How to Land a Product Manager Job in Technology*. CareerCup.
- McGilchrist, S. (2011). Music 'releases mood-enhancing chemical in the brain'. <https://bbc.in/3pnDd3j>.
- Meisler, B. (2012). The real reason silicon valley coders write bad software. *The Atlantic*.

- Miles, J., Welby, S., Moura, J., Jones, R., Berg, B., Lavington, S., Ibbett, R., Denning, P., and Furber, S. (2022). IEEE milestone award event: University of manchester june 21st 2022. <https://youtu.be/zfqYCVUqXfQ>.
- Miller, C. (2019). *The Death of the Gods: The New Global Power Grab*. Windmill Books.
- Minds, S. (2016). Why is it important to talk about student mental health? <https://youtu.be/PiqcSSDwvvY>.
- Mitchum, R. (2019). I introduced the term 'dad-rock' to the world. i have regrets. <https://www.esquire.com/entertainment/music/a29419783/what-is-dad-rock/>.
- Moore, D. (2019). Libraries gave us power: A letter to llandrindod, from medellin. <https://www.iwa.wales/agenda/2019/02/libraries-gave-us-power-a-letter-to-lindrindod-from-medellin/>.
- Morali, J. and Willis, V. (1978). Y.M.C.A. In Morali, J., editor, *Cruisin'*. Village People, Casablanca Records.
- Mulqueeny, E. (2011). Year 8 is too late. <https://mulqueeny.wordpress.com/2011/08/10/year-8-is-too-late/>.
- Murashenkov, D. (2019). What are some common criticisms of cracking the coding interview? <https://www.quora.com/What-are-some-common-criticisms-of-Cracking-the-Coding-interview>.
- Newton, I. (1675). Letter to Robert Hooke. *Historical Society of Pennsylvania*.
- Nurse, P. (2017). Don't work too hard! <https://youtu.be/vI0DyPkz9JI>.
- Nye, C. (2021). Being proud of going to state school. <https://bbc.in/3A45MYq>.
- Octocat (2020). Mastering issues. <https://guides.github.com/features/issues/>.
- Ong, J. and Lopez, L. (2019). Create your resume for google: Tips and advice. <https://youtu.be/BYUy1yvjHxE>.
- O'Shaughnessy, A. (1874). Ode. <https://www.poetryfoundation.org/poems/54933/ode->.
- Ozuzu, J. (2020). Massive thanks to Morgan Stanley for putting my picture up in Times Square, New York to commemorate my start with the firm! https://www.linkedin.com/posts/jason-ozuzu-a87049173_mymssummer-morganstanley-msinterns2020-activity-6690233743011139584-Yq_C.
- O'Toole, G. (2012). If I Had More Time, I Would Have Written a Shorter Letter. <https://quoteinvestigator.com/2012/04/28/shorter-letter/>.
- Parker, P. (1962). With great power comes great responsibility. https://en.wikipedia.org/wiki/With_great_power_comes_great_responsibility.

- Peroni, S. (2021). Computational thinking and programming. <https://comp-think.github.io>.
- Petroski, H. (1992). *To Engineer Is Human: The Role of Failure in Successful Design*. Vintage.
- Phillips, J. and McKenzie, S. (1967). San Francisco (Be Sure to Wear Flowers in Your Hair). In Adler, L., editor, *The Voice of Scott McKenzie*. Monterey International Pop Music Festival, Ode Records.
- Pluchino, A., Biondo, A. E., and Rapisarda, A. (2018). Talent versus Luck: The role of randomness in success and failure. *Advances in Complex Systems*, 21(03n04):1850014.
- Pointer, A., Pointer, J., and Pointer, R. (1984). I'm so excited. In Christian, C., editor, *So Excited!* The Pointer Sisters, Planet Records.
- Potter, J. A., Press, M. C., Callaghan, T. V., and Lee, J. A. (1995). Growth responses of Polytrichum commune and Hylocomium splendens to simulated environmental change in the sub-arctic. *New Phytologist*, 131(4):533–541.
- Poundstone, W. (2013). *Are You Smart Enough to Work at Google?* Oneworld Publications.
- Price, H. (2019). What is sexual harassment at work? how to tell when lines are crossed in the workplace. <https://bbc.in/38JSge7>.
- Pritchard, C. (2012). Fergie time: Does it really exist? <https://bbc.in/3Puj3PV>.
- Práxedes, C. F. (2020). Running a hackathon in lockdown. <http://man.ac.uk/Dqo94I>.
- Práxedes, C. F. (2022). My industrial placement at the walt disney company. <http://man.ac.uk/5StpJw>.
- Raymond, E. S. (1999). *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media.
- Redmond, P. (2010). *The Graduate Jobs Formula: How to land your dream career*. Crimson Publishing.
- Reich, R., Sahami, M., and Weinstein, J. M. (2021). *System Error: Where Big Tech Went Wrong and How We Can Reboot*. Harper Collins.
- Richards, S. (2019a). Content design london: Readability guidelines, treat links with respect. <https://readabilityguidelines.co.uk/content-design/links/>.
- Richards, S. (2019b). Why people do not read online. <https://readabilityguidelines.co.uk/content-design/how-people-read/>.
- Ridenhour, C., Sadler, E., Boxley, H., and Boxley, K. (1989). Fight the power. In Lee, S., editor, *Do the Right Thing*. Public Enemy, The Bomb Squad.

- Ridenhour, C., Shocklee, H., Sadler, E., and Drayton, W. (1988). Don't believe the hype. In Squad, T. B., editor, *It Takes a Nation of Millions to Hold Us Back*. Public Enemy, Def Jam.
- Ritchie, D. M. (1993). The development of the C language. *ACM SIGPLAN Notices*, 28(3):201–208.
- Ro, C. (2021). How overwork is literally killing us. <https://bbc.in/3wcBotN>.
- Robbins, A. (2012). *The Geeks Shall Inherit the Earth: Popularity, Quirk Theory, and Why Outsiders Thrive*. Hachette.
- Roberts, R. J. (2015). Ten Simple Rules to Win a Nobel Prize. *PLOS Computational Biology*, 11(4):e1004084.
- Robertson, R. (2020). Fixing a bug on my open source project: From start to finish. <https://dev.to/rose/fixing-a-bug-on-my-open-source-project-from-start-to-finish-1749>.
- Rogers, J. (2008). Dad rock anyone? <https://www.theguardian.com/music/2008/may/16/popandrock.paulweller>.
- Ronson, M. and Mars, B. (2014). Uptown funk. In Bhasker, J., editor, *Uptown Special*. Mark Ronson, Columbia Records.
- Rose, A. (1992). November rain. In Clink, M., editor, *Use your illusion*. Guns N' Roses, Geffen Records.
- Ross, J. (2021). The biggest companies in the world in 2021. <https://www.visualcapitalist.com/the-biggest-companies-in-the-world-in-2021/>.
- Rowling, J. K. (1997). *Harry Potter and the Philosopher's Stone*. Bloomsbury.
- Ruben, A. (2017). The postdoc: A special kind of hell. *Science*.
- Ryder, S. (1988). Wrote for luck. In Hannett, M., editor, *W.F.L. Think About the Future Mix*. Happy Mondays, Factory Records. album.
- Ryder, S. (2019). *Wrote for Luck: Selected Lyrics*. Faber and Faber.
- Saini, A. (2018). *Inferior: How Science Got Women Wrong and the New Research That's Rewriting the Story*. Harper Collins.
- Saini, A. (2019). *Superior: The Return of Race Science*. Harper Collins.
- Sandwell, I. (2020). Educating greater manchester: What happened to drew povey? harrop fold school is now under different leadership. <https://www.digitalspy.com/tv/a34573273/educating-greater-manchester-drew-povey-resignation-explained/>.
- Santos, L. (2020). Five favorite coping tips for world mental health day. <https://youtu.be/kCQim9VrnDY>.

- Santos, L. (2022a). Burnout and how to avoid it: Sometimes our careers can take us over and make us miserable. <https://www.happinesslab.fm/2022-new-year-mini-season/burnout-and-how-to-avoid-it>.
- Santos, L. (2022b). The science of well-being at yale university. <https://www.coursera.org/learn/the-science-of-well-being>.
- Schellmann, H. (2022). Finding it hard to get a new job? robot recruiters might be to blame. <https://www.theguardian.com/us-news/2022/may/11/artificial-intelligence-job-applications-screen-robot-recruiters>.
- Schuh, P. (2004). *Integrating Agile Development In The Real World*. Charles River Media, Inc., Rockland, MA, USA, 1st edition.
- Scott, R. (1974). Boy on a bike. <https://player.bfi.org.uk/free/film/watch-hovis-boy-on-the-bike-1974-online>.
- Sedgwick, R. (2019). Should computer science be required?
- Servant, A. (2020a). Discrimination: your rights and protected characteristics. <https://www.gov.uk/discrimination-your-rights>.
- Servant, A. (2020b). Handing in your notice. <https://www.gov.uk/handing-in-your-notice>.
- Servant, A. (2021a). Disciplinary procedures and action against you at work. <https://www.gov.uk/disciplinary-procedures-and-action-at-work>.
- Servant, A. (2021b). Dismissal: your rights. <https://www.gov.uk/dismissal>.
- Servant, A. (2021c). Raise a grievance at work. <https://www.gov.uk/raise-grievance-at-work>.
- Shaha, A. (2014). Wave machine demonstration. https://youtu.be/VE520z_ugcU.
- Shakespeare, W. (1597). *Romeo and Juliet*.
- Shapiro, B. R., Lovegall, E., Meng, A., Borenstein, J., and Zegura, E. (2021). Using role-play to scale the integration of ethics across the computer science curriculum. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. ACM.
- Shaw, D. (2021). Duolingo boss: 'develop your social skills'. <https://bbc.in/3QqMwLI>.
- Shelley, P. B. (1812). Declaration of rights. In Murray, E. B., editor, *The Prose Works of Percy Bysshe Shelley, Volume 1*. Romantic Poetry, Oxford University Press.
- Shields, J. (2019). What happens to resume tables and columns in an ats? see for yourself. <https://www.jobscan.co/blog/resume-tables-columns-ats/>.

- Shimer, D. (2018). Yale's most popular class ever: Happiness. <https://nyti.ms/3cpxAud>.
- Sinatra, F. (1969). My way. In François, C., editor, *My Way*. The Mafia, Reprise Records.
- Sissay, L. (2015). Inspire and be Inspired. <https://youtu.be/WzZs1w3NWzg>.
- Smith, M. E. and Hanley, S. (1986). Lucifer over Lancashire. In Leckie, J., editor, *Bend Sinister*. The Fall, Beggars Banquet.
- Smith, M. E., Smith, B., and Rogers, S. (1987). Hit the north. In *The Frenz Experiment*. The Fall, Beggars Banquet.
- Smith, M.-S. (2021). Graduate salaries in the UK. <https://luminate.prospects.ac.uk/graduate-salaries-in-the-uk>.
- Smith, S. and Rosser, K. (2021). Understanding the impact of job scams. <https://www.gov.uk/government/news/understanding-the-impact-of-job-scams>.
- Snow, C. P. (1959). *The Two Cultures and the Scientific Revolution*. Cambridge University Press.
- Somers, J. (2011). How I failed, failed, and finally succeeded at learning how to code. *The Atlantic*.
- Soo, S. (2022). Rotten mango: true crime + all things spooky podcast. <https://www.rottenmangopodcast.com>.
- Spearpoint, A., Hewison, D., Crawford, J., Saunders, P., and Wilson-Wright, I. (1992). It's not what you know. In Leon, C., editor, *Body Exit Mind*. New Fast Automatic Daffodils, Play it again Sam.
- Spielberg, S. (2002). Minority report. [https://en.wikipedia.org/wiki/Minority_Report_\(film\)](https://en.wikipedia.org/wiki/Minority_Report_(film)).
- Spinellis, D. (2021). Why computing students should contribute to open source software projects. *Communications of the ACM*, 64(7):36–38.
- Spärck-Jones, K. and Runciman, B. (2007). Computing is too important to be left to men. *ITNOW*, 49(4):18–20.
- Stanford-Clark, A. (2019). Innovation begins at home. <http://man.ac.uk/ZyTpe0>.
- Stanley, A. (2022). The big picture on wealth, income, ecological, and gender inequality looks bad. *IMF: Financy and Development*.
- Stoke, H. and Green, V. (2013). *A Dictionary of Bristle*. Behringer, 4th edition.
- Stuart, M. (1971). Willy wonka & the chocolate factory. https://en.wikipedia.org/wiki/Willy_Wonka_%26_the_Chocolate_Factory.
- Stumpe, B. (2014). The ‘touch screen’ revolution. In *From Physics to Daily Life*, pages 103–116. Wiley-VCH Verlag GmbH & Co. KGaA.

- Styles, H., Hull, T., and Johnson, T. (2022). As it was. In Harpoon, K., editor, *Harry's House*. Harry Styles, Columbia Records.
- Swinton, J. (2019). *Alan Turing's Manchester*. Infang Publishing.
- Sylor-Miller, K. and Evans, J. (2021). Oh shit, git! recipes for gitting out of a git mess. <https://wizardzines.com/zines/oh-shit-git/>.
- Taylor, R. (1984). Radio ga ga. In Mack, R., editor, *The Works*. Queen, EMI Records.
- Tenant, N. and Lowe, C. (1986). Opportunities (let's make lots of money). In Jeczalik, J. J., editor, *Please*. Pet Shop Boys, Parlophone.
- Thresher, B., Ticker, D., Wriggles, B., Body, K., McCartney, B., and Seal, R. (2007). Chippy tea. In Eckerslike, W., editor, *Never Mind the Hotpots*. The Lancashire Hotpots, Fuss Records.
- Thunberg, G. (2019). Our house is on fire: Davos speech urging economists to tackle runaway global warming. <https://www.theguardian.com/environment/2019/jan/25/our-house-is-on-fire-greta-thunberg16-urges-leaders-to-act-on-climate>.
- Tickle, L. (2015). Where are all the young school governors? under 30s aren't signing up to governance in the same way they want to be environmental campaigners or trek for charity. <https://www.theguardian.com/teacher-network/2015/mar/11/young-people-school-governors>.
- togetherall (2020). A quick look inside togetherall. <https://youtu.be/mXEbpau9YjU>.
- Toliver, C., Oshunrinde, E., Gomringer, K., and Gomringer, T. (2019). No idea. In Wondagurl and Cubeatz, editors, *Heaven or Hell*. Don Tolliver, Cactus Jack Records.
- Tupper, H. and Ellis, S. (2020). *The Squiggly Career: Ditch the Ladder, Embrace Opportunity and Carve Your Own Path Through the Squiggly World of Work*. Portfolio Penguin.
- Tupper, H. and Ellis, S. (2021). Squiggly careers and the end of the traditional career path. https://www.ted.com/talks/sarah_ellis_helen_tupper_squigglycareers_and_the_end_of_the_traditional_path.
- Turing, A. (1950). Programmers' handbook for manchester electronic computer mark ii. *Computing Machine Laboratory, University of Manchester*.
- Tyldum, M. (2014). The imitation game.
- Verkaik, R. (2019). *Posh Boys: How English Public Schools Ruin Britain*. Oneworld Publications.

- Verkaik, R. (2021). A bullingdon in reverse: how working-class student club is taking on elitism. <https://www.theguardian.com/education/2021/mar/06/bullingdon-in-reverse-working-class-student-93-club-taking-on-elitism>.
- Wachowski, L. and Wachowski, A. (1999). The Matrix. https://en.wikipedia.org/wiki/The_Matrix.
- Waititi, T. (2022). Thor: Love and thunder. https://en.wikipedia.org/wiki/Thor:_Love_and_Thunder.
- Walker, D. and Walker, R. (1992). *Energy, Plants and Man*. Oxygraphics Ltd; 2 edition.
- Walker, M. (2018). *Why We Sleep: The New Science of Sleep and Dreams*. Penguin Books.
- Ware, B. (2011). *Top Five Regrets of the Dying*. Hay House UK Ltd.
- Ware, B. (2017). The top five regrets of the dying. <https://youtu.be/nayz3xJxRTA>
- Wax, R. (2014a). Mindfulness part 1: Dodging the bullets. https://youtu.be/7TCCLY_lV6M
- Wax, R. (2014b). *Sane New World*. Hodder.
- Weber, L. (2012). Your résumé vs. oblivion: Inundated companies resort to software to sift job applications for right skills. <https://on.wsj.com/3pqbAqw>.
- Wickham, H. and Grolemund, G. (2017). *R for Data Science*. O'Reilly UK Ltd.
- Wilby, P. (2011). Cheltenham ladies' college: 'this isn't a pink, frilly school'. <https://www.theguardian.com/education/2011/aug/01/cheltenham-ladies-college-principal-interview>.
- Williams, R. and Chambers, G. (1997). Let me entertain you. In Power, S., editor, *Life thru a Lens*. Robbie Williams, Chrysalis Records.
- Wodiany, I. (2018). Interns in technology salaries 2018 report. <https://github.com/IgWod/interns-salaries-uk-2018>.
- Woodruff, W. (2003). *Beyond Nab End*. Abacus.
- Woolston, C. (2022). Why science needs more research software engineers: Ten years after their profession got its name, research software engineers seek to swell their ranks. *Nature*.
- Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2 edition.
- Xie, Y. (2017). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida.
- Xie, Y., Dervieux, C., and Riederer, E. (2020). *R Markdown Cookbook*. Chapman and Hall/CRC, Boca Raton, Florida.

- Yaden, D. B., Claydon, J., Bathgate, M., Platt, B., and Santos, L. R. (2021). Teaching well-being at scale: An intervention study. *PLOS ONE*, 16(4):e0249193.
- Zatorre, R. J., Chen, J. L., and Penhune, V. B. (2007). When the brain plays music: auditory-motor interactions in music perception and production. *Nature Reviews Neuroscience*, 8(7):547–558.