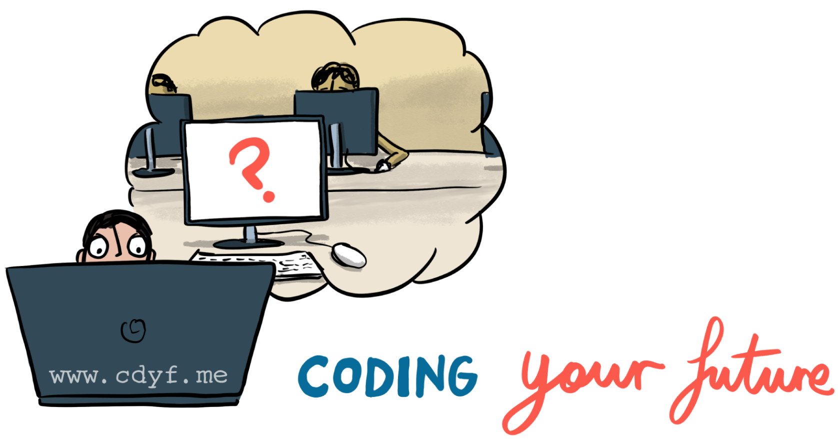
Coding Your Future: A Guidebook for Students

Duncan Hull at the University of Manchester and illustrated by Bryan Mathers at Visual Thinker

Contents

Welcome to your future



Hello and welcome to *Coding Your Future*, the guidebook that will help you to design, build, test, debug and code your future in computing. This guidebook is aimed at students in higher education, both those studying Computer Science as part of their degree or those from other disciplines with an interest in computing.

This guidebook (also available as cdyf.pdf and cdyf.epub) supports second year teaching at the University of Manchester, but it DOES NOT MATTER:

- 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 *subject* you are studying, provided that you are computationally curious
- what *institution* you are studying at, this book is University and institution agnostic
- *where* in the world you are studying

So there is probably something in this guidebook for *any* student of computing.

0.1 Visualising your future

A lot of self-help literature can be dry, dull, textbooky and boring with few illustrations and conversations.

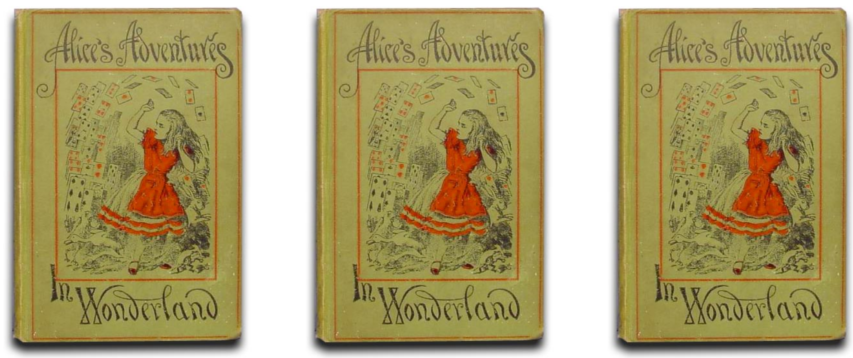


Figure 1: The cover of the 1898 edition of the novel *Alice's Adventures in Wonderland*. Public domain image via Wikimedia Commons w.wiki/327E

In the novel *Alice's Adventures in Wonderland* (Caroll, 1865), the heroine Alice is looking at book her sister is reading:

> 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?" – See gutenberg.org (Caroll, 1865)

So this book uses pictures and conversations wherever possible to help you understand and visualise your future.

0.2 Your future aims

This guidebook aims to help you develop stronger habits of mind, body and soul using five key ingredients:

1. **Code:** Instructions contained in this guidebook

2. **Data:** Facts, statistics and images collected together for your analysis
3. **You:** Activities for you to do in addition to reading
4. **Futures:** Possible futures for you to think about. Try not to dwell on the past. Think about the future. (Ryder, 1988, 2019) Think about *your* future.
5. **Me:** Hello, my name is Duncan. I'm your tour guide here. If you're feeling a bit lost, follow me.

Figure 2: Hello my name is Duncan. If you're feeling a bit lost, follow me. Image adapted from "Hello my name is sticker" by Eviatar Bach, public domain w.wiki/32RV

Coding your future explores techniques for making career decisions, job searching, 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 tackles professional 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

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 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 a job search strategy and identify employers, sectors and roles that are of interest to you

3. Improve your written communication skills both for job applications and communicating with other people
4. Plan and prepare competitive written applications using standard techniques including CVs, covering letters, application forms and digital profiles
5. Compete successfully in interviews and assessment centres by preparing for technical and non-technical questions
6. Plan further steps in your career such as promotion, postgraduate study & research, alternative employment and longer term goals
7. 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 prerequisites

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.

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

Don't read this book, gut it! Reading this book from cover to cover like a novel is not recommended. That would be foolish.

Figure 3: Don't read this book, gut it like a fish. Gutting fish in Isla Margarita image by Wilfredor via Wikimedia commons w.wiki/_23m

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!" (Woodruff, 2003)

So, gut this book like a fish. 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 three parts. The first part is on designing your future while the second is on building and testing your future shown in the map in figure 4. The final part is a help section, for rebooting your future. Let's look in a bit more detail at the content of each of the three parts of this guidebook:

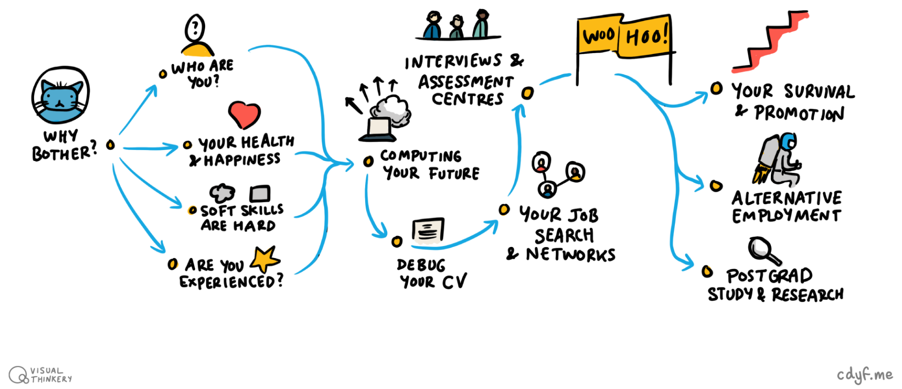


Figure 4: Mapping your future: Each yellow dot on this diagram is a chapter in *Coding Your Future*. The chapters on the left tackle design issues like *who are you*? Chapters on the right tackle the practicalities of executing and testing your career choices, such as *debugging your CV*. Mapping your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

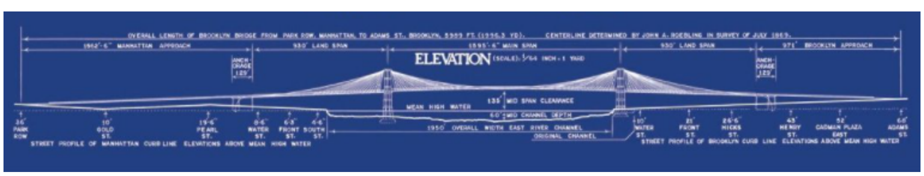


Figure 5: Designing your future is about drawing up a blueprint, like this one for the elevation of the Brooklyn Bridge in New York. What does your blueprint look like? The chapters on designing your future will help you fill in the details.

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

1. Investigating your future looks at why you should bother reading any of this guidebook
2. Knowing your future challenges you to reflect on who you are, what makes you unique and why you are here
3. Nurturing your future encourages you to take care of your mental and physical health
4. Writing your future explores your soft skills, and how they complement your hard skills and why employers value them so much
5. Experiencing your future asks you to reflect on your work experience and help identify where you can improve it
6. Computing your future looks at the role computing can play in your career, with or without a Computer Science degree

0.5.2 Building and testing your future

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

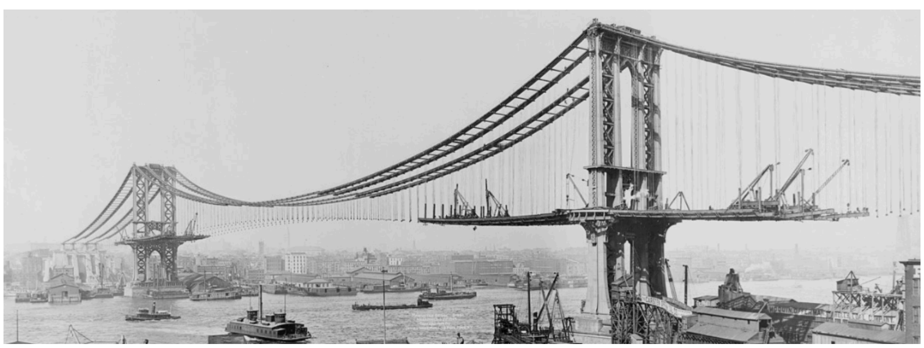


Figure 6: Just like the Manhattan Bridge, your future will be easier to build once you've done some design. You don't need a grand design with tonnes of details, a simple sketch will do. Design questions are covered in the first part of this guidebook: designing your future. Picture of the Manhattan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

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.

1. Debugging your future looks at debugging your written communication such as covering letters, application forms and digital portfolios.

2. Finding your future, looks at where and how can you look for interesting opportunities
3. Speaking your future, looks how can you turn interviews to your advantage and negotiate any offers you receive
4. Surviving your future looks at the next steps. Once you've landed a job, how will you survive and thrive outside (and after) University
5. Researching your future discusses if a Masters degree or a PhD right for you?
6. Broadening your future encourages you to broaden your horizons. Maybe you want to start your own business and employ others or you'd like to work in the non-profit or public sector? Perhaps you could be a freelancer or contractor? The possibilities are endless.

0.5.3 Supporting your future

The third part of this book, contains supporting material that will help the design, build and test phases described above. You'll need good support to help with the stresses and strains of building your future shown in 7

Figure 7: The Clifton Suspension Bridge in Bristol has supporting chains which can move when heavy loads pass over the Avon valley bridge. You'll need good support to cope with the stresses and strains of building your future. Clifton suspension bridge picture adapted from original by Nic Trott via Wikimedia commons w.wiki/32tu

1. Ruling your future provides *Ten Simple Rules for Coding your Future*, this book in a nutshell
2. Reading their future invites you to put yourself in the employers shoes by reading some CVs
3. Moving your future looks at opportunities outside of capital cities like London
4. Hearing your future invites you to listen to students stories of their transition from education to employment
5. Actioning your future gets you to think about your actions by emphasising verbs on your job applications

6. Scheduling your future invites you to set aside an hour each week for tackling the issues described in this book
7. Reading your future everything we've cited in this guidebook, because books are good for your soul

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:

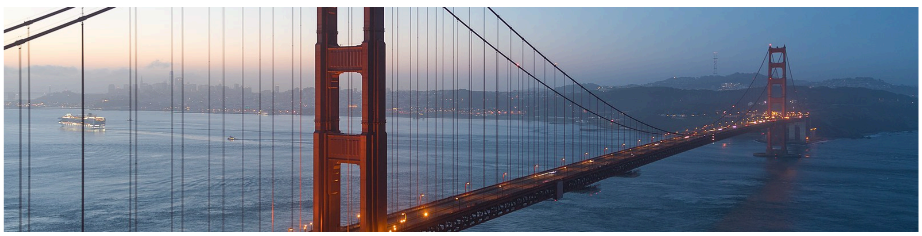


Figure 8: The iconic Golden Gate Bridge in California, adapted from an original picture by Frank Schulenburg (CC BY-SA) on Wikimedia Commons w.wiki/37kY

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
5. **Discussing** your future: breakpoints invite you to stop your code from executing and think about the variables and parameters you are using. Can they be improved? Reflect and discuss.
6. **Reading** your future: reading stuff because its 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 (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 Acknowledgements

The content of this book is based on hundreds of conversations I have had with students of (mainly) Computer Science and Maths with some Physics and Engineering, since 2012. I've also spoken to many of their employers too.

0.7.1 Standing on the shoulders of students

First and foremost, I'd like to thank all the students who have helped with this book, both directly and indirectly.

> "If I have seen further it is by standing on the shoulders of students."
> (Newton, 1675)

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. It has been my pleasure and privilege to work with you all.

Figure 9: 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

I'd especially like to thank 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 Luke Beamish and Petia Davidova. In addition, the PASS leaders and facilitators, 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! If you sent me a CV and I didn't reply, I apologise. There are limits to what is humanly possible. The chapters on debugging your future (self assessment) and reading their future (peer assessment) are based on the most common bugs (or are they features?) I've seen in CVs.

So, thank you students for being studious.

0.7.2 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. (Googler, 2019)

So, thanks employers for employing our students.

0.7.3 Thank you colleagues

I've also had significant support from colleagues in the Department of Computer Science (@csmcr) and support staff at the University of Manchester. (@ManUniCareers, @alumniUoM, @OfficialUoM)

I would especially like to thank Jim Miles for encouraging me to write a book shortly after he offered me a job. I thought he was joking (about the book) but it actually turned out to be another one of Jim's great ideas. Thanks Jim.

I'd also like to thank the only three people in the whole world who've had the misfortune of reading all of my PhD thesis; 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 arse (to write this book) in the Midland Hotel, Manchester at the May ball.

Thanks to Carole Goble for re-teaching me how to write by covering drafts of my MSc thesis in red ink (and swear words [1]) and Steve Furber for playing bass in our "boy band" Tuning Complete.

0.7.3.1 Thanks to academic staff

Thanks to past and present academic colleagues, PhD students and teachers at the University of Manchester who have contributed to this guidebook and the environment it was written in. We are bound together by the power of weak ties alongside stronger forces and friendships. They include (in alphabetical order):

[1] the swear words didn't appear until the sixth of seventh draft

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, Lynne Bianchi, Helena Björn van Praagh, Stewart Blakeway, Petrut Bogdan, Caroline Bowsher, Linda Brackenbury, Judy Brewer, Nick Brown, Mihai Bujanca, Oscar Corcho, Christian Brenninkmeijer, Andy Bridge, Andy Brass, Andy Brown, Gavin Brown, Terry V. Callaghan, Grant Campbell, Angelo Cangelosi, Peter Capon, Andy Carpenter, Nicola Carrier, Barry Cheetham, Ke Chen, Sarah Clinch, Ian Cottam, Brian Cox, Simone Di Cola, Paul Dobson, Clare Dixon, Danny Dresner, Nick Drummond, Warwick Dunn, Doug Edwards, Iliada Eleftheriou, Anas Elhaig, Suzanne Embury, Michael Emes, Alvaro Fernandes, Jonathan Ferns, Michele Filannino, Nick Filer, Paul Fisher, Steve Furber, Andre Freitas, Aphrodite Galata, Matthew Gamble, Jim Garside, Kristian Garza, Chris Gilbert, Danielle George, Richard Giordano, Birte Glimm, Carole Goble, Rafael Gonçalves, Antoon Goderis, Roy Goodacre, Bernardo Cuenca Grau, Peter R. Green, Keith Gull, John Gurd, Luke Hakes, Robert Haines, Guy Hanke, Simon Harper, Phil Harris, Jonathan Heathcote, Lloyd Henning, Gareth Henshall, Andrew Horn, Farid Kahn, Matthew Horridge, Ian Horrocks, Toby Howard, Roger Hubbold, Luigi Iannone, Jane Ilsley, Jules Irenge, Daniel Jameson, Caroline Jay, Mirantha Jayathilaka, 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 Kotsiopoulos, Oliver Kutz, Alice Larkin, Peter Lammich, John Latham, Kung-Kiu Lau, Margi Lennartsson Turner, Dave Lester, Peter Li, Zewen Liu, Phil Lord, Mikel Luján, Darren Lunn, Matthew Makin, Nicolas Matentzoglu, Paul Mativenga, Erica McAlister, April McMahon, Merc and members of the Manchester University Mountaineering Club (MUMC), Simon Merrywest, Eleni Mikroyannidi, Colin Morris, Norman Morrison, Georgina Moulton, Boris Motik, Christoforos Moutafis, Tingting Mu, Ettore Murabito, Mustafa Mustafa, Javier Navaridas, Kostas Nikolou, Aleksandra Nenadic, Goran Nenadic, 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 Orihuela, Stuart Owen, Ali Owrak, Pavlos Petoumenos, Luis Plana, Jackie Potter, Malcolm Press, Colin Puleston, Paul Nutter, Ignazio Palmisano, Dario Panada, Michael Parkin, Bijan Parsia, Jon Parkinson, Norman Paton, Jeff Pepper, Steve Pettifer, Rishi Ramgolam, Allan Ramsay, Alasdair Rawsthorne, Farshid Rayhan, Alan Rector, Giles Reger, Graham Riley, David Robertson, Jeremy Rodgers, Clare Roebuck, Jeremy Rodgers, 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, Jonathan Shapiro, Vangelis Simeonidis, Liz Sheffield, Bushra Sikander, Kieran Smallbone, Alastair Smith, Stian Soiland-Reyes, Irena Spasic, David Spendlove, Robert Stevens, Alan Stokes, Shoaib Sufi, James Sumner, Peter Sutton, Neil Swainston, John H. Tallis, Paul Taplin, Federico Tavella, Chris Taylor, Tom Thomson, Dave Thorne, David Toluhi, Tony

Trinci, Dimitri Tsarkov, Daniele Turi, Jake Vasilakes, Laura Vasques, Delia Vazquez, Giles Velarde, Chiara Del Vescovo, Markel Vigo, Andrei Voronkov, Niels Walet, Alex Walker, Louise Walker, Dieter Wiechart, Igor Wodiany, Katy Wolstencroft, Natalie Wood, Chris Wroe, Crystal Wu, Lisheng Wu, Yifan Xu, Viktor Yarmolenko, Yeliz Yesilada, Serafeim Zanikolas, Xiao-Jun Zeng, Jun Zhao, Liping Zhao, Ning Zhang and Evgeny Zolin.

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 cynicism shown in figure 10, I am an optimist when it comes to the power of natural languages.

Figure 10: Christopher Hitchens explains the difference between autobiography and memoir (Hitchens, 1997)

0.7.3.2 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, Cassie Barlow, Jennie Ball-Foster, Emma Bentley, Christine Bowers, Karen Butterworth, Chris Connolly, Ellie Crompton, Jean Davison, Gavin Donald, Miriam Cadney, Chris Calland, Ben Carter, Hannah Cousins, Holly Dewsnip, Tammy Goldfeld, Penney Gordon-Lanes, Amelia Graham, Iain Hart, Kath Hopkins, Lynn Howarth, Yvonne Hung, Susie Hymas, Radina Ivanova, Alex 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, Melanie Price, Chris Rhodes, Graham Richardson, Martin Ross, Julian Skyrme, Elaine Sheehan, Angela Standish, Martine Storey, Bernard Strutt, Jannine Thomas, Joseph Tirone, Daisy Towers, 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, thank you colleagues for being collegiate. You all make the University of Manchester a *fantastic* place to work, even during a global pandemic.

0.7.4 Thank you scholars

Beyond Manchester there is a wider academic community of scholars that have influenced 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.
- Thanks to Quintin Cutts, Steven Bradley and Jane Waite for helping me setup and run SIGCSE journal club. Thanks to all the journal clubbers too, many of our journal club conversations have fed directly into the content of this guidebook
- Thanks to David Malan (@malan) for CS50 which continues to be 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, 2021) which was been a significant influence on this book had a gradual but dramatic effect on my personal and professional life
- 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 Athene Donald at Occams Typewriter and Stephen Curry at Reciprocal Space for writing entertaining and inspiring blogs
- Thanks to Jessica Wade for inspiration, support and educating me on some of the issues of equality, diversity and inclusion, especially gender and race.

- Thanks to Jonathan Black (@JonathanPBlack) for his book *Where am I Going, Can I Have a Map?*, (Black, 2017) his *Financial Times* columns (Black, 2019a) and videos.
- Thanks to David Alan Walker for his book *Energy, Plants & Man* which inspired the conversations and pictures idea behind this book.

So thanks scholars for being scholarly.

0.7.5 Thank you Bath

As a graduate of the Postgraduate Certificate in Education (PGCE) in Science at the University of Bath (graduated 2011), I have been heavily influenced by the fantastic work of PGCE science course leaders Caroline Padley (Physics), Steve Cooper (Chemistry), Malcolm Ingram (Biology) and fellow students on the course.

Figure 11: Panorama of the world heritage site and Georgian City of Bath, Somerset. Image cropped from an original by David Iliff available under CC BY-SA 3.0 license via Wikimedia Commons at w.wiki/32BS

Thanks Bath for the initial teacher training (ITT), the medicinal Aquae Sulis and the beautiful Cotwolds Area of Outstanding Natural Beauty (AONB).

0.7.6 Thank you Shaftesbury

Thanks to Stuart Ferguson, David Booth, Chris Almond, David Ball, Caroline Dallimore, Mr Travers, Caroline Moss and all the other staff and students at Shaftesbury School who hosted my first PGCE teaching placement. Thanks also to my fellow Shaftesbury/Bath trainees Katharine Platt, Harriet Edwards, Vicky Dury and Joan Shaw for sharing their hard won knowledge through peer learning. Thanks Joan for keeping me awake on the long and winding west country highways 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)

Figure 12: Shaftesbury is the home of Gold Hill and Shaftesbury School. Image of Gold Hill by Sean Davis via Wikimedia Commons w.wiki/32gw made with the Wikipedia app.

0.7.7 Thank you Swindon

Thanks to headteacher 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 waves using Alom Shaha's jelly babies and kebab sticks shown in figure 13.

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

0.7.8 Thank you schools

Thanks to all the schools who interviewed (but rejected 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. Although unsuccessful, these interviews were very productive failures:

- Wrightlington School, Radstock, Somerset, see their amazing Orchid project wsbeorchids.org run by Simon Pugh-Jones
- The Cooper School, Bicester, Oxfordshire, see their teacher in my pocket project

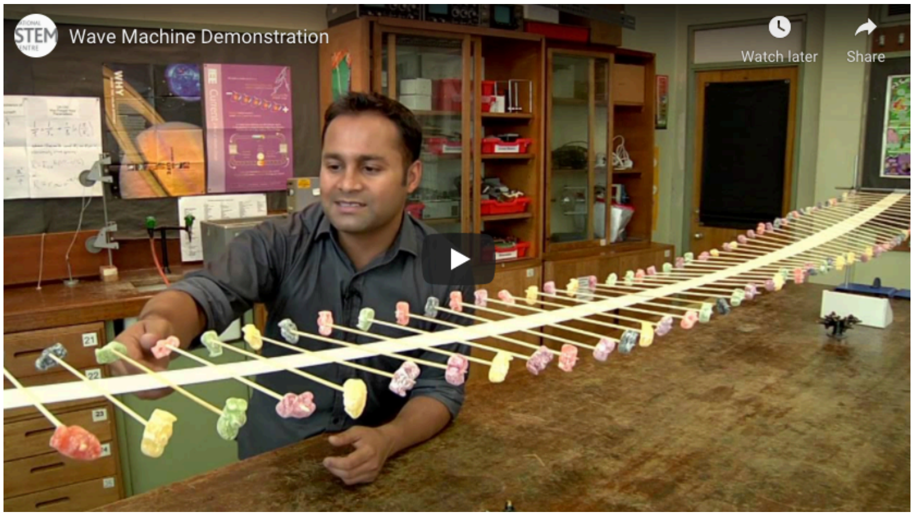


Figure 13: A wave machine demonstration by Alom Shaha with Physics and jelly babies. What's not to like? (Shaha, 2014)

- St John's Marlborough, Wiltshire - not to be confused its posher and more famous next door neighbour Marlborough College
- Oasis Academy, Brislington, Bristol
- Redland Green School, Redland, Bristol
- The John of Gaunt School, Trowbridge, Wiltshire
- Didcot Girls' School, Didcot, Oxfordshire
- Cheltenham Ladies' College, Cheltenham, Gloucestershire[2]
- Blackburn College, Lancashire "I read the news today, oh boy! Four thousand holes in Blackburn, Lancashire" (Lennon and McCartney, 1967)

So thanks schools, for schooling.

0.7.9 Thank you Stockport

Thanks to headteacher Joanne Meredith, her team of staff and the students at St. Annes R.C. High School, Stockport for hosting my Newly Qualified Teacher (NQT) year. Thanks to Keith Doran and other members of the *alternative staff room* for your emotional, moral and practical support throughout the year. According to the *Manchester Evening News*, St. Anne's is "the forgotten school" (Johnson, 2020; Gill and Statham, 2021), see figure 14, but I'll never forget you or the lessons you taught me.

[2]As a newly trained Jedi knight, freshly armed with a PGCE, I was anxious for my first teaching job and momentarily considered using my pedagogical powers on the "dark side" of the force: private education. (Green and Kynaston, 2019) Forgive me for I have sinned!

Figure 14: Good governance is crucial to good schools. Many schools like St. Anne's, and the hundreds of children 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? Take a look at governorsforschools.org.uk. Fair use image via Wikimedia Commons w.wiki/33Xs made with the Wikipedia app

So thanks Stockport for being Stockport, the magnificent Stockport Viaduct and for The Hatters! It's all that matters, Stockport Hatters.

0.7.10 Thank you Moravians

Thanks to Thsespal 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 learnt heaps as visiting supply teachers of English and Mathematics, thanks to an introduction from a mutual contact Angus Barker.

Figure 15: 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 of the Moravian Church in Ladakh, Jammu & Kashmir state to provide education for Tibetan refugees fleeing from their homeland across the Himalayas.

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

0.7.11 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 her 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) for bookdown.org, the software used to produce this book alongwith the comprehensive and well-written documentation on using it. [Xie (2017); Xie (2015); Xie et al. (2020);]
- Thanks to Bronnie Ware for her *The Top Five Regrets of the Dying* (Ware, 2011) which helped me to re-align my priorities when they were all out of kilter

- Thanks to blokes on the interwebs whose words I've enjoyed reading including Tim Bray at ongoing, Paul Downey at whatfettle.com, Paul Graham at paulgraham.com, Peter Norvig at norvig.com and Neil Saunders at What you're doing is rather desperate. Your writing is existence proof that engineers and scientists should also be good communicators.
- Thanks to Sophie Milliken for *From Learner to Earner: A recruitment insider's guide for students wanting to achieve graduate job success* (Milliken, 2019) which draws useful distinctions between graduate jobs and graduate schemes

So, thanks influencers for being influential.

0.7.12 Thank you githubbers

Thanks to everyone who has contributed via github. I will credit *any* github contributors here, small or large. Even the typos, it all counts. You can easily add yourself to this roll call by correcting my delibreate mitsakes.

Keith Mitchell (@apiadventures), Jan Machacek (@janm399), Zee Somji (@ezeethg), Tsvetankov (@Tsvetankov), teobalmos (@teobalmos)

If you'd like to contribute via github you can:

- raise an issue at github.com/dullhunk/cdyf/issues/new
- click on the `edit this page` on any page at cdyf.me which will initiate a pull request
- `git clone https://github.com/dullhunk/cdyf.git` the repository to submit pull requests from your setup
- submit a pull request github.com/dullhunk/cdyf/pulls

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

0.7.13 Thank you Bryan

Many of the illustrations for this book have been drawn by the very talented Bryan Mathers @BryanMMathers shown in figure 16.

Bryan is an artist, visual thinker and entrepreneur, who also happens to have a Bachelors degree in Computer Science from the University of Glasgow. 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 *sooo* glad we randomly bumped into each other at a conference shown in figure 8.3.

Figure 16: Bryan Mathers Self portrait by Visual Thinkery is licensed under CC-BY-ND

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

0.7.14 Thank you friends

Thanks to my friends, especially those who I've enjoyed singing, dancing and live music with. I hope we can sing and dance together to live music again before too long.

So, thank you friends for your friendship.

0.7.15 Thank you family

To my mum, dad, brother, sister, wife, son, , 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 for your unconditional love. Σ .

0.8 About me

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

So what's *my* story? Like many people, my path has been what Helen Tupper and Sarah Ellis call a "squiggly career" rather than classic linear one. (Tupper and Ellis, 2020) I've been gainfully employed as a paperboy, supermarket cashier, shelf stacker, sausage factory worker, pork pie filler, chef, dogsbody, field assistant, database administrator, deli counter server, consultant, match-day steward, envelope stuffer, high school teacher, postdoc, research scientist, software engineer, lecturer, external examiner, tutor and academic.

I've done a range of voluntary work too, serving as a competition judge, fundraiser, rabble rouser, code club & coderdojo leader, digital council member, school governor, curator, librarian, beer drinker, wikipedia trainer, journal clubber and editor. But as Ronnie Lane and Ronnie Wood once said:

> "I wish that I knew what I know now, when I was younger." — Ronnie Lane (Lane and Wood, 1979)

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 useful. I've sat on both sides of the interview table, as interviewer and interviewee. I have had some spectacular failures, alongside some modest successes, and have included personal stories where they are relevant.

Most of what I have learned about employment comes from listening to, and watching students interact with employers as they take the first tentative steps in their careers. I've documented some of what they taught me, so reading this book may help you learn from some of their successes and failures.

0.9 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 and therefore do not represent University policy. Also:

0.9.1 Licensing

This guidebook is published under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License (CC-BY-NC-ND) license. 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
- You do not make any derivative works

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

Images in this guidebook are published under different licenses, depending on the source. For example, Bryan Mathers illustrations are licensed CC-BY-ND, others are different. Each figure caption gives details.

0.9.2 Your privacy

This site is hosted on netlify.com, see the netlify privacy policy. This site also uses netlify analytics to monitor website use which complies with the General Data Protection Regulation (GDPR).

Some of these services use cookies. These can be disabled in your browser, see 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.

Part I

DESIGNING YOUR FUTURE

Chapter 1

Investigating your future

The first half of this book is about designing your future. So before we get started, let's tackle a fundamental design issue. Why the hell would you want to bother investigating your future? Why should you bother with reading this pesky guidebook when you have so many other things on your plate 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!

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

- Identify and decide which parts of this book you are going to read
- Set your expectations for using this guidebook
- Travel down the rabbit hole into the underworld of work

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* (Caroll, 1865), the heroine 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 come down here anyway.

1.3 Your future is your responsibility

When Andy Stanford-Clark started working at IBM, fresh out of University, his boss gave him the following advice:

> "Nobody cares about your career except you." —Anon (Stanford-Clark, 2019)

Andy is now Master Inventor and Chief Technology Officer (CTO) at IBM in the UK so it was probably good advice. Another, slightly more positive way of putting the advice is, the person who cares *most* about your career is you. So while there are 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, academic staff, friends, your careers service, employers and your wider network but ultimately it is *your* responsibility to sort out whatever comes next. That might sound obvious but don't wait for somebody else to do it for you, because it probably won't happen.

1.4 Your degree is not enough

You've worked incredibly 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 whatever it is you are studying at University.

Under these circumstances, you might be tempted to believe that the world owes you something in return for your hard work. 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 isn't going to set you apart much from your competition. Even having a first class degree (Coughlan, 2019a; Borrett, 2019) may not distinguish you that much your competitors. 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)

What **will** distinguish you from your competitors will be your experience, your projects, your communication skills and any awards or honours you might have picked up along the way. 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. The increase in the number of graduate schemes and graduate jobs has not kept pace with this growth in graduates which means that each graduate job or graduate scheme has more and more graduates applying for it
3. There are lots of graduates in your discipline, for example around 9,000 every year in Computer Science alone in the UK. What makes you different from the other 8,999 computer scientists graduating in your year?

> 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 a degree can open doors to many careers in computing. What the data in Figure 1.2 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, but it is typically not enough by itself.

1.5 Maximising your future

Studying at University is a significant investment of your time and money. 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:

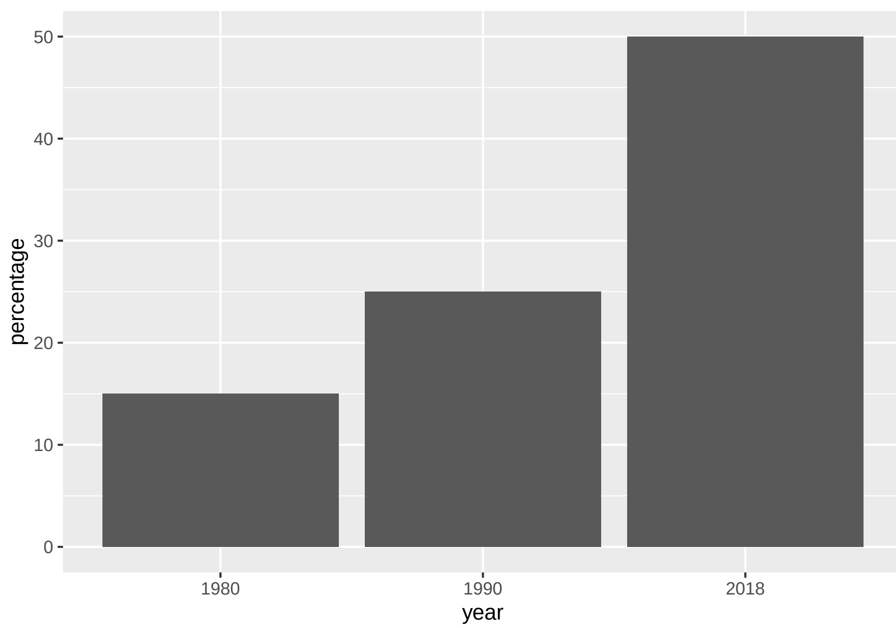


Figure 1.2: Percentage of young people in the UK going into higher education between 1980 and 2018. Over the last forty years, the proportion of young people going into higher education has more than doubled from 15% in 1980 to over 50% in 2018. Data taken from BBC news article on the symbolic target of 50% at university reached (Coughlan, 2019b)

1. Help you maximise the return on the substantial investment of time and money (ROI) you've put into your study
2. Give you an overview of important professional issues that are sometimes neglected or sidelined in 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.

1.6 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. Procrastination is a part of the human condition (Graham, 2005):

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

It probably doesn't help that many of 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. Broadly speaking, the professional issues described in this book are usually covered by pastoral support systems, counselling services, careers services, trade organisations, professional bodies, student unions and their societies,.

Despite being sidelined, these issues matter and it is in your own selfish 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 start thinking about them and doing something. The sooner you start thinking about them the better decisions you'll make about what comes next. It's 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, and sometimes knowing what you **don't** want to do is just as valuable as knowing what you do.

Computer scientists call this problem "search space reduction", (Ferrari et al., 2008) because 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 because it's too late when you graduate.

1.7 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

Examination	What examiners are assessing	Grade
CV, application form covering letter	• Should we invite you to interview ? • Can you communicate well in writing?	pass/fail
Interview	• Should we offer you a job? • Can you communicate well verbally? • Can you communicate well nonverbally?	pass/fail
Employee performance	• Should we promote you? • Should we give you a pay rise? • Should we extend your contract?	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 Resume can escape, we'll say more about this in chapter on debugging your CV.

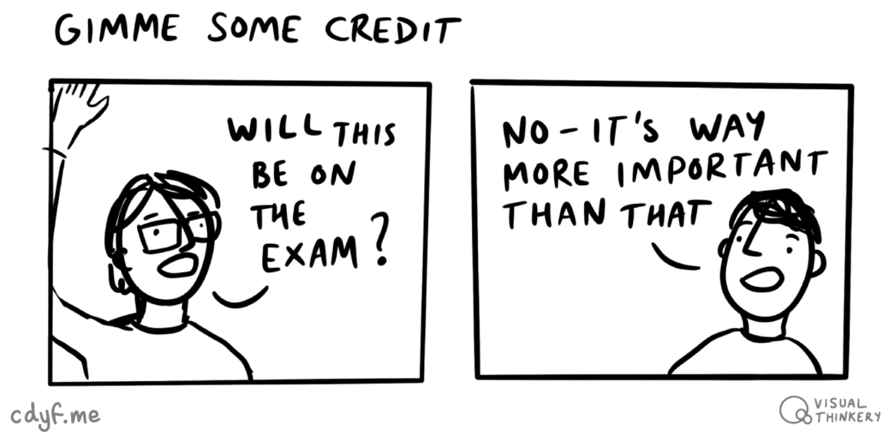


Figure 1.3: *no* this will not be on the exam set by University, but *yes* it will be on the exams set by employers. Some of the most important exams you sit at (and after) University are set by employers. This guidebook will help you prepare for those exams and increase your chances of passing them. Gimme some credit figure by Visual Thinkery is licensed under CC-BY-ND

It's a similar story with interviews, if you fluffed and interview question or came across badly, it can be really difficult to find out from the employer what you did wrong.

1.8 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. Just reading a guidebook won't get you very far, you'll need to do the activities in this book to get the most out of it.

1.9 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.

Figure 1.4: There are tonnes of resources out there offering advice on the huge range of professional issues. You can't read them all, but this guide will help you navigate the resources that will be most interesting and useful to you

Lots of professional advice is readily available, but how will you navigate it? This book signposts you 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.

1.10 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 several ways

- View the entire book source on github at github.com/dullhunk/cdyf
- Click the **Edit this page**, which appears on the right hand side of every chapter with the octocat logo
- Raise an issue github.com/dullhunk/cdyf/issues/new
- Submit a pull request github.com/dullhunk/cdyf/pulls, If you're not familiar with pull requests, see makeapullrequest.com
- Email me suggestions for improvements if you don't want to use github

Any corrections or suggestions that get included will be fully acknowledged in the acknowledgements, unless you tell me otherwise. We welcome all improvements, however small.

1.11 Your future is different

I wrote this guidebook because I needed a resource for students to help them design, build, test and debug their futures. I wanted a single resource that could help students compete for jobs while at University, or shortly after graduating. I could not find anything suitable that met all the requirements of the students I was teaching. So I wrote this one which contains some new material and recommends the best resources if you want to know more. These are found in the signposts sections of each chapter.

This book aims to combine these perspectives and to be different from existing resources in the following ways:

1.11.1 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.

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

1. Check and verify any facts and claims I make in this book for yourself
2. Go and consult the original sources if you think they might be useful

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 section. The University of Manchester has physical and electronic copies of many (but sadly not all) of the books listed here.

We're 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.

1.11.2 Guiding your future

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 embark on the first stages of their careers.

Although this is a course guidebook used in the second year undergraduate teaching, you don't need to be enrolled on the course to benefit from reading it, watching the videos and doing the exercises and coding challenges.

1.11.3 Your future is constantly updated

You are reading the alpha version, the Minimum Viable Product (MVP) of this guidebook. 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 podcasts in the Coding your Future
- More illustrations throughout the book
- Improved content, finish incomplete chapters
- Fix bugs and typos
- Your suggestions for improvements and corrections, via github etc

I'm taking a Release Early, Release Often (Raymond, 1999) approach to publishing this guidebook, you could call it agile book development. (Schuh, 2004)

> Agile: make it up as you go along. Waterfall: make it up before you start, live with the consequences.
>
> — Paul Downey (@psd) February 19, 2015

1.11.4 I'm deliberately writing in first person narrative

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 stories to illustrate key points.

1.11.5 Your future has no paywall

You don't need to pay anything to read this book online because its openly available, see the license terms (CC-BY-NC-ND). Publishing this guidebook online makes it findable and accessible, something that isn't true of lots 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).

1.11.6 Your future has audio & video

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

1.12 Engaging 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 it can be improved, let me know. I always welcome constructive feedback, especially when it comes via a pull request.

> Let's go Captain pic.twitter.com/LN33dh5dip
>
> — VeraM (@FonikhSoupia) July 22, 2020

1.13 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 elses. There are lots of people 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 in surviving your future, so *The Clean Coder* is an essential signpost for chapter 10 as well.

1.14 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 (including this one) has a TL;DR summary.

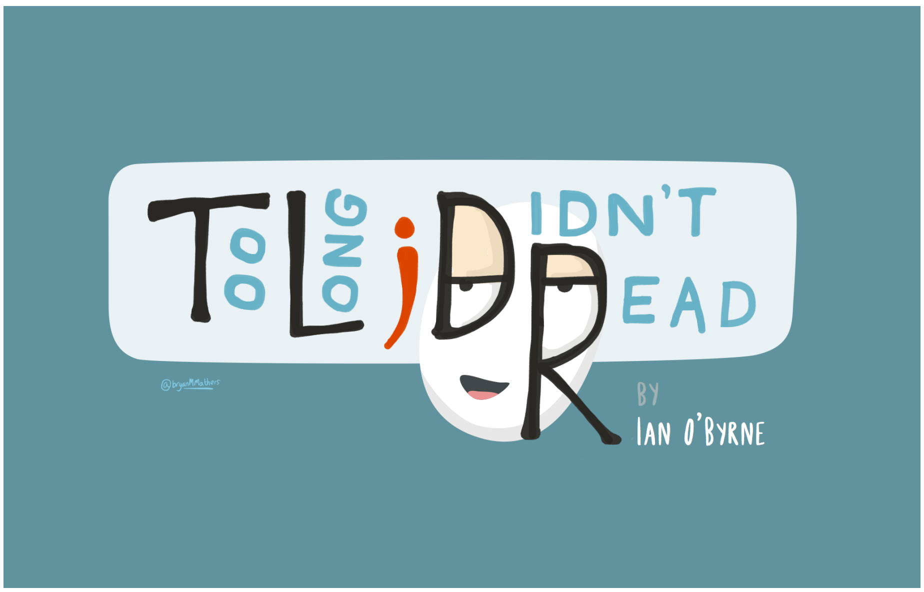


Figure 1.5: Too long, didn't read? TL;DR by Visual Thinkery is licensed under CC-BY-ND

The TL;DR for this chapter is, you should read this guidebook because it will help you design, build, test, debug and code your future in computing.

Chapter 2

Knowing your future

Hello, who are you? What's your story? 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 the best possible version of you? Knowing your possible futures depends on knowing who you are now.

2.1 What you will learn

Reading this chapter and doing the activities will help you to

- Improve your self-awareness, knowing yourself better will help you to know your future more clearly
- Describe what you know: what is in your **head**,
- Describe what have you done: (**hands**)
- outlining some of your values: what is in your **heart**
- Identify your protected characteristics
- Check and be grateful for any privileges that you may have

2.2 What's your story, coding glory?

> "If you are human, you love stories. Why? We're hardwired to love stories because they help us understand our world and are essential to our evolution. We use stories to organise and communicate our surroundings and our past, present and future" —Heather Box and Julian Mocine-McQueen (Box and Mocine-McQueen, 2019)

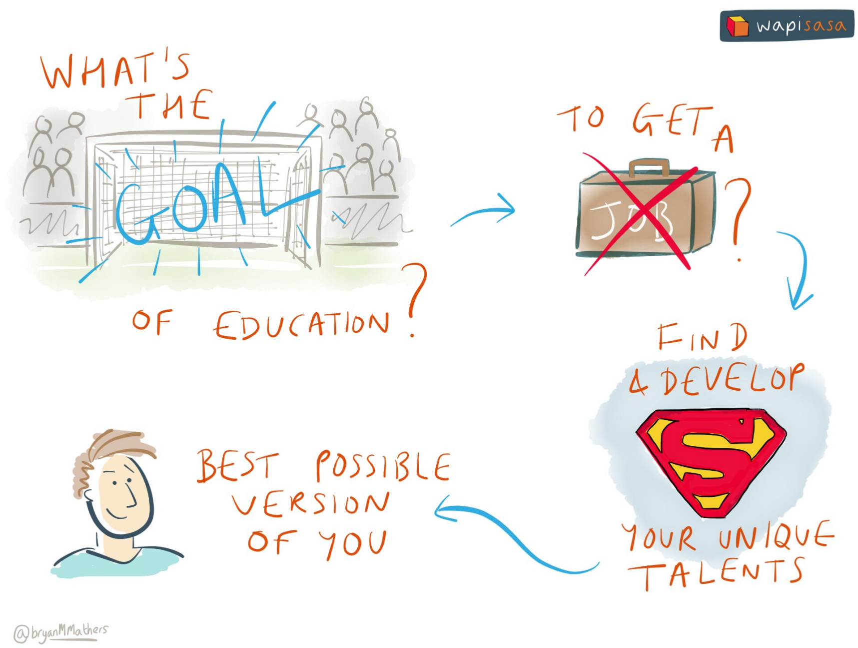


Figure 2.1: Your education is a crucial part of your story and who you are. The purpose of your education is not just to get you a job but to find and develop your unique talents. What are you unique talents? How are you developing them? Goal of Education sketch by Visual Thinkery is licensed under CC-BY-ND

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 self-awareness is to think about what your story is. (Box and Mocine-McQueen, 2019) How did you get here, where are you going, what has inspired you? Who is the authentic you? (Ware, 2011) What are your hopes and dreams? By starting to answer these questions you will gain a better understanding of who you are. This includes strengths, weaknesses, motivation and values. (Bolles, 2019)

Universities offer many opportunities for self improvement, self discovery and developing your unique skills. One way to build your self-awareness is to reflect on your knowledge, values and skills. In Waldorf education this is characterised as "head, heart and hands". (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 in the future?

Answering these questions will help you understand your story.

2.3 Ikigai: What is the meaning of life?

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 realise. You can develop better self-awareness by describing four attributes shown in Figure 2.2, together these are known as your ikigai ( ) 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 shown Figure 2.2. In practice, you may realistically only be able to achieve one, two or three. That said, it's still a valuable exercise to think about what is in each category for you.

2.4 Self assess your ikigai

Take a sheet of paper, draw the four overlapping rings shown in Figure 2.2, and spend five to ten minutes adding things in each ring.

Figure 2.2: Reasons for being, a concept in Japanese known as “ikigai”. Image by Emmy van Deurzen on Wikimedia Commons. According to ikigai, a meaningful life combines doing four things. 1. What you are good at 2. What you love 3. What the world needs and 4. What you can get paid for. Illustration by Nimbosa derived from works in the public domain by Dennis Bodor and Emmy van Deurzen, CC BY-SA 4.0 on Wikimedia Commons w.wiki/qWT

- 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?

Figure 2.3: How well do you know yourself. Know who you are sketch by Visual Thinkery is licensed under CC-BY-ND

Thinking about your ikigai will clarify your knowledge of yourself. Some parts of your identity are so important that they are protected by legislation, in the UK and in other countries. The next section looks at those.

2.5 Your protected characteristics

Some of your characteristics are protected. The Equality Act of 2010[1] protects you from discrimination at work or in education, based on what are known as "protected characteristics". (UK, 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

[1] http://www.legislation.gov.uk/ukpga/2010/15/contents

- 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)

2.6 Coding challenges

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?

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

2.7 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

2.7.1 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 peoples regrets that will illuminate your own values and might just change your life. 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 2.4.

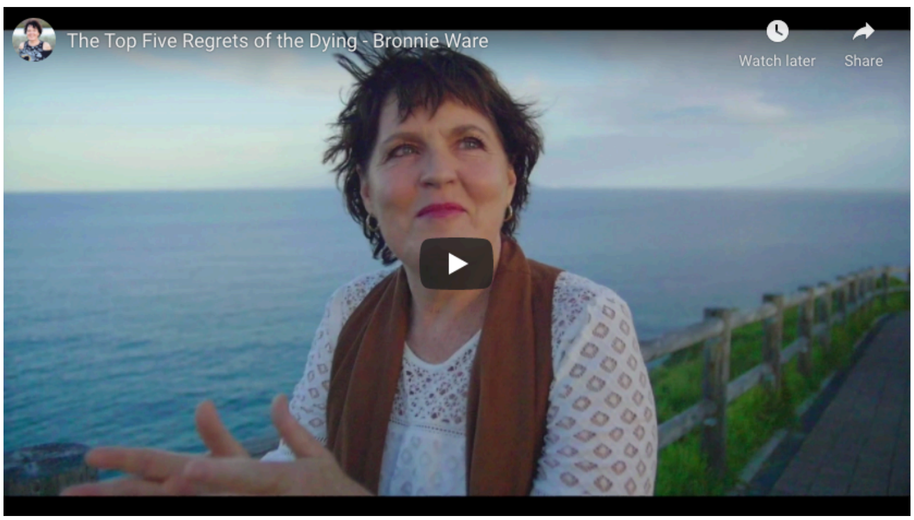


Figure 2.4: Palliative care nurse Bronnie Ware explains the top five regrets of the dying. [@youtube-bronnie] Bronnie learned a lot from looking after people on their deathbeds, then wrote it all down in a fantastic book [@regrets].

2.7.2 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 he explains how you can't possibly decide what to do in five years time in the video in figure 2.5. Where do you see yourself in five years time? is a question some interviewers like to ask.

Figure 2.5: Where will you be five years from now? Best-selling author Dick Bolles talks at the Googleplex about the gaps between education and employment. [@youtube-bolles]

2.7.3 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.

2.7.4 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 the chapter on your well-being So:

- **If you're 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're 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're 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 educated in Britain (or elsewhere) you should read *Engines of Privilege: Britain's Private School Problem* (Green and Kynaston, 2019)
- **If you're 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're 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

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

2.8 Summarising self awareness

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

This chapter has looked at who you are. Being self aware, understanding your strengths and weaknesses is key to getting what you want from your career. 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 need to keep thinking about your identity because some aspects of your identity may be constantly changing.

These are fundamental design questions you'll need to address when you starting building your future. We touched on understanding any privileges you may have as being important for understanding who you are but also in being beneficial for your mental health.

In the next chapter, we'll look at mental health in more detail.

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 nurturing and caring for yourself now will also nurture your future.

3.1 What you will learn

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

- Identify some of the symptoms of mental ill health in yourself and your peers, particularly anxiety and depression
- Describe five self-help techniques for improving mental health
- 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
- Schedule activities for improving mental and physical health into your daily or weekly routine
- DISCLAIMER: I am neither a medical doctor or a psychologist: If you're affected by mental ill health, you should speak to a trained professional. This chapter just gives you a quick overview of mental health and points you to where you can find out more.

Figure 3.1: Alan Turing was an outstanding Computer Scientist, but did you know he was also a respectable athlete too? He ran, cycled and rowed to relieve stress, (Kottke, 2018) and came close to competing in the Olympics as a runner. 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.

3.2 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) On describing his work for the government and why he punished himself so much in training, Alan Turing said:

> "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. (Kottke, 2018)

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 and an ongoing pandemic of COVID-19 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

The *persistent* and *lasting a long time* are important here because while its part of the human condition to worry and feel low, that doesn't *necessarily* mean you are affected by poor mental health.

3.2.1 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 3.2 and at 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.

3.2.2 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

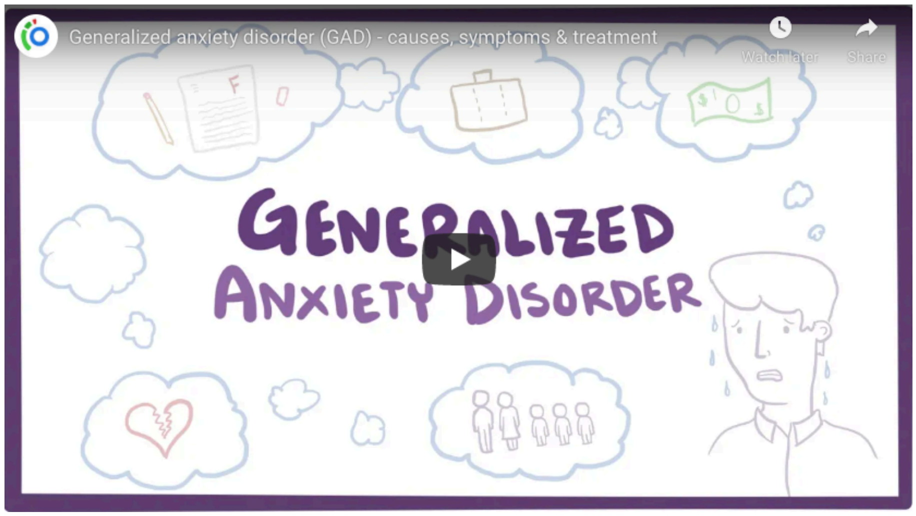


Figure 3.2: Generalised anxiety disorder is a condition characterised by excessive, persistent and unreasonable amounts of anxiety and worry about everyday things. (Desai, 2016) Note that the video takes an American perspective using American terminology such as DSM–5.

illustrator Matthew Johnstone tells the story of how he overcame the "black dog of depression" in the video in Figure 3.3 made in collaboration with the World Health Organization (WHO).

3.2.3 Drugs

Prescription medication can help some people with their mental health. For example, when I was affected by depression, Selective Serotonin Reuptake Inhibitors (SSRIs) worked for me, shown in Figure 3.4, but they don't for everybody. Sometimes the drugs don't work, they just make you worse. (Ashcroft, 1997)

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 often 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 at the end of this chapter (Santos, 2021)

Figure 3.3: Matthew Johnstone explains how he overcame the affects of depression, using the metaphor of the black dog (Johnstone, 2012)

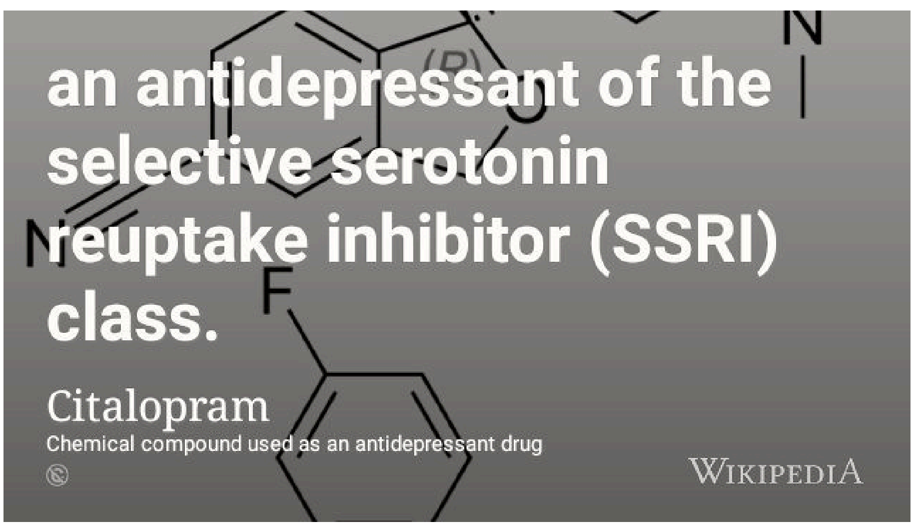


Figure 3.4: Citalopram is a type of antidepressant known as a Selective Serotonin Reuptake Inhibitor (SSRI). SSRI's can help some people who have been affected by depression. They work for some people (including me) but they don't for everybody. Skeletal formulae of Citalopram by Vaccinationist via Wikimedia Commons w.wiki/3Ddn adapted using the Wikipedia app.

3.3 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 3.5: 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 3.6. 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 (either small or big)
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 the signposts section below

4. **Socialising**: prioritise time with friends and family, rather than turning inward or diving deeper into work
5. **Mindfulness**: be mindful of emotions using the R.A.I.N. technique:

   - **R**ecognise: negative emotions
   - **A**ccept: accept emotions rather than fighting them
   - **I**nvestigate: notice how the emotion feels inside your body
   - **N**urture: 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 3.6.

Figure 3.6: Laurie Santos describes five coping techniques for improving well-being: Exercise, gratitude, sleep, getting social and meditation (Santos, 2020).

So there are things you can do to help yourself, but you may also need to seek help from others.

Sometimes a desire to be productive by working hard has the opposite effect, because the sacrifices you make can be counter-productive.

> pic.twitter.com/D2SP4iJspT
>
> — lizandmollie (@lizandmollie) February 23, 2020

3.4 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, its 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:

3.4.1 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.

3.4.2 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

3.4.3 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 3.7.

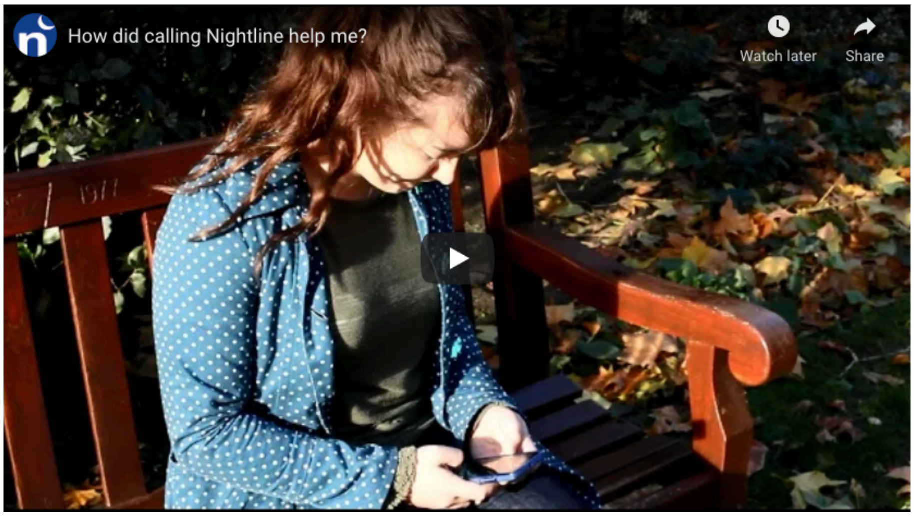


Figure 3.7: Students explain in their own words how calling Nightline helped them whilst at university. [@youtube-nightline]

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.

3.4.4 The Samaritans

The Samaritans are a charity who provide emotional support to anyone in the 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.

3.4.5 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, 2014) She is interviewed by Students Against Depression in the video in figure 3.8 about using mindfulness to "dodge the bullets" of depression.

Figure 3.8: 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. [@youtube-wax]

3.4.6 Papryus

Suicide is the biggest killer of under 35's in the UK. 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.

3.4.7 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.

3.4.8 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 3.9

Figure 3.9: Stephen Fry, President of Mind, describes how MIND tackles misconceptions around mental health and social stigmas. [@youtube-we-are-mind]

3.4.9 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 3.10 which describes why its important to talk about student mental health.

3.4.10 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 3.11 which describes why its important to talk about student mental health.

Figure 3.10: Talking about mental health is a crucial part of helping those who are suffering from it [@youtube-student-minds]

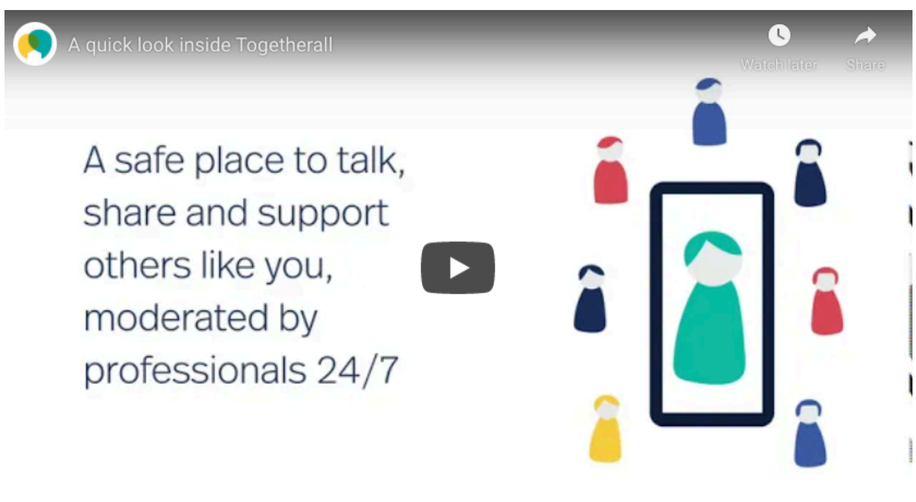


Figure 3.11: A quick look inside togetherall, an online community for people who are stressed, anxious or feeling low. [@youtube-togetherall]

3.5 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 we describe in the chapter on searching 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 applications 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.

Figure 3.12: A fixed mindset is monolithic like the Easter island statues, known as 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 3.12. If your grades aren't has 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 3.13. 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 can you do better next time.

According to Stanford psychologist Carol Dweck we can all be placed on a spectrum according to 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 to suggest 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 3.13: Psychologist Carol Dweck explains the power of "not yet" and the growth mindset (Dweck, 2014)

3.6 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 do 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 3.14

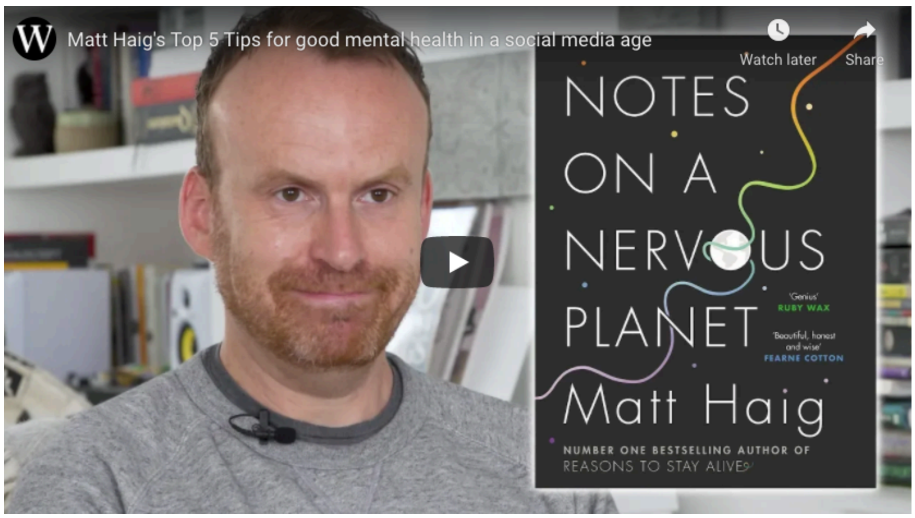


Figure 3.14: Two of Matt Haig's top five tips for good mental health (Haig, 2018) include 1. Being more careful (and mindful) of social media and 2. Reading more books because books are good for your soul. Not just his book. Any book. Books are good for you. Trust me on this. (Forever, 2019)

There's plenty of evidence to suggest 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 3.15.

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-beig. (Santos, 2021) 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

Figure 3.15: Social media like LinkedIn clearly has its uses (see debugging your links) but you 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) You don't have to be on social media by andy-leek.com photographed by Duncan Cumming

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 3.6 gives you a brief taster of Laurie's style and work.

3.7 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.

If you are affected by mental ill health, talking about it is the first place to start, but often the hardest. 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.

Take my thoughts with you and when you look behind, you will surely see, a face that you recognise, you're not alone. (Kellett et al., 1997)

Chapter 4

Writing your future

Let's get straight to the point of this chapter: your soft 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, and 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 deploying your softer skills.

Figure 4.1: 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. Sorry to break the bad news! This means you need to constantly improve your softer skills and provide evidence of them to potential employers. Other people sketch by Visual Thinkery is licensed under CC-BY-ND

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)

So 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, we *all* find communication difficult, and everyone can get better at it. This chapter takes a look at the softer skills and techniques you can use to improve your communication with other people, whatever mood they are in and whosever team they are on.

4.1 What you will learn

- Recognise the importance of written communication, both as a reader and a writer
- Identify examples of where written communication is crucial in science and engineering
- Improve your written communication skills using some simple writing and reading exercises
- Identify the importance of teamwork

4.2 Computing is your superpower!

Studying computer science gives you an awesome superpower. We will look at some of the reasons why in the chapter 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.

Your computational superpower is less powerful if it isn't complemented by a broad range of softer skills. Typically, these skills are not emphasised (by repeated assessment) in most computer science degrees. This not because soft skills aren't important but because they are hard to measure accurately.

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 doing this. However, if I want to know how good you are at using

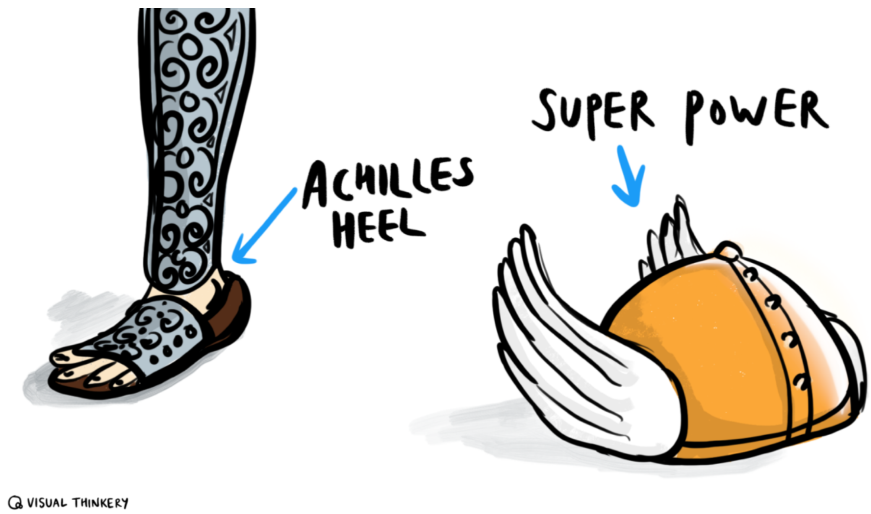


Figure 4.2: Computing is a superpower that gods like Hermes and mortal heroes like Achilles would probably have envied. (Fry, 2018) As a technical or "hard skill", computing is a crucial weapon in your armoury but what are your weaker skills? What is your Achilles' heel? For some scientists and engineers, their weakness is their soft skills, such as communication and team work. This chapter looks at what you can do to improve them 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 communication skills to work in a team, negotiate, lead, resolve conflicts, persuade others, show empathy etc that's **much** harder to measure accurately.

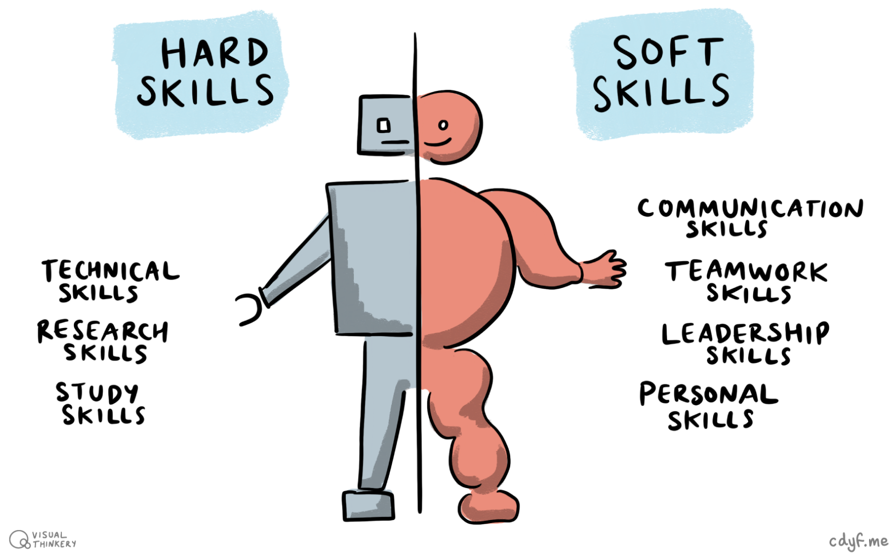


Figure 4.3: Hard skills and soft skills aren't much use without each other. You will need both to survive and thrive but most science and engineering education focuses on your hard skills, not your soft skills. Why? Because hard skills are often much easier to measure. Hard and soft skills sketch by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

4.3 Communication I/0

In terms of input and output, your fundamental communication skills are listening, speaking, reading and writing words in natural languages shown in table 4.1. These are the "assembly languages" of human communication. This might sound blindingly obvious, but these skills are often under-estimated or undervalued 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.

Engineers and scientists sometimes lack communication skills outlined in table 4.1. Think of your stereotypical scientist, clad in a white coat, unable to explain the complexities of their subject to people inside their lab, let alone outside of it. Then there is the nerdy software engineer stereotype who prefers the

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

company of computers to people. Yes, these are lazy and sometimes unhelpful stereotypes, but they express public perception of scientists and engineers as poor communicators.

4.3.1 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 written communication skills 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 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 understandable 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 good documentation, 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. However, I'm arguing that these careers wouldn't be needed if software engineers were better at documenting, explaining and communicating with other human beings their code 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 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.

4.3.2 Natural language engineering

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: LinkenIn, Facebook, WhatsApp, Twitter, blogs, Medium.com etc
- bug reports and messages in issue trackers like Jira, BugZilla and Trello
- 'How to' and cookbook style articles and books
- API reference material
- in-code documentation `# comments in code`
- Self-documenting code
- Executable specifications in test suites like cucumber.io
- Laboratory manuals and laboratory notebooks
- The one page summary for management
- User documentation, release notes
- Case studies of software use
- Frequently Asked Questions (FAQ)
- 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
- bookdown.org mixes code and natural language

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.

4.4 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?

4.4.1 Try Google's Tech Writing course

Google have developed some excellent technical writing courses including:

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

These courses run as part of the second year course COMP2CARS at the University of Manchester, see the course schedule 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.

4.4.2 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:

- private diary
- gratitude journal
- public web log or blog
- personal notes gathered somewhere (e.g. private github repository)
- 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 the wellbeing chapter) to gardening. (Leendertz, 2006)

4.4.3 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 the references. Find a time and place to do this every day and stick to it.

4.4.4 Dogfooding

Some companies test their products by trialling them on their own employees, this is sometimes known as eating your own dogfood. Tasty, tasty dogfood.

You can use a similar approach to testing your own writing, known simply as **Dogfooding**. Let's say you've just written a covering letter. 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. There's something about articulating words out loud that flushes out errors you don't pick up when you read them. This 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 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.

4.4.5 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 Manual. RTFM. Read THIS Friendly Manual, see figure 4.4

4.5 Coding challenges

- Write an article or blog post about something you care about, find a suitable venue for publication

Figure 4.4: As well as learning from other people's hard won experience, reading the friendly manual (RTFM) will also improve you written communication skills. Just like you improve your coding skills by writing and reading code, improve your written communication skills by reading and writing in natural langauges like English. RTFM poster by Jeremy Keith, modified by Atlaslowa is licensed under CC-BY on Wikimedia Commons w.wiki/vBX

- 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.

4.6 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, motivation and leadership are other soft skills that are important too. How can you develop these skills while at University? How can you demonstrate to potential employers that you have these skills?

This chapter is under construction because I'm using agile space station development methods, see figure 4.5.

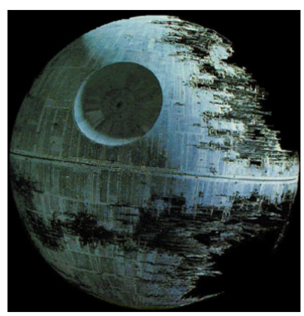


Figure 4.5: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

Chapter 5

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?

5.1 What you will learn

By the end of this chapter you will be able to

- Describe why having experience can improve your chances of getting interviews
- Identify what counts as experience and why it's valuable
- Recognise opportunities to get more experience before you graduate

5.2 Why is experience so valuable?

It's common for students to be focused on their grades, whether those grades are low, middling or 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. Either way, you are *much more* than your grades, because your education is only a part of who you are, shown in in figure 5.3. You are the sum total of your experiences, this is one of the reasons that experience is so valuable.

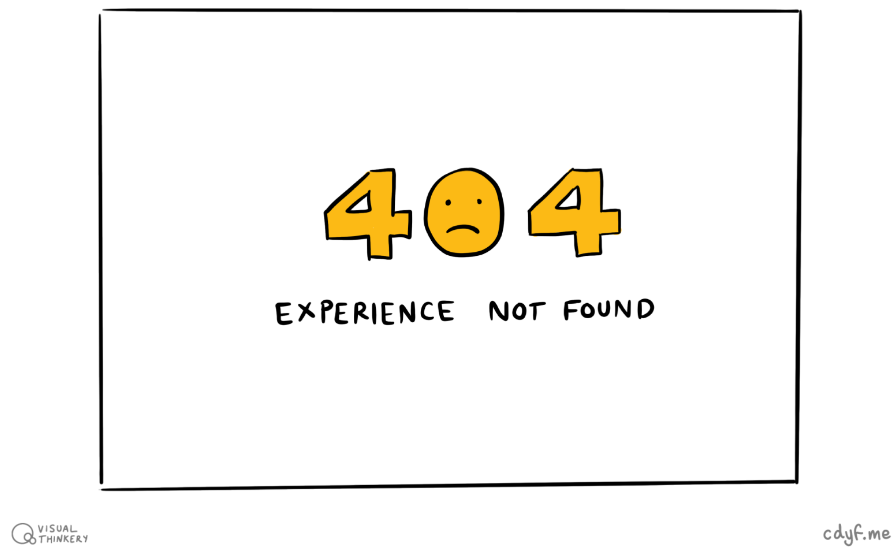


Figure 5.1: Do you respond with a sheepish *experience not found* error message when people ask about your experience? Is your experience like the classic HTTP 404 page not found? The client sent you a valid request for your experience, but your server couldn't find it. Awkward. Embarrassing silence? 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

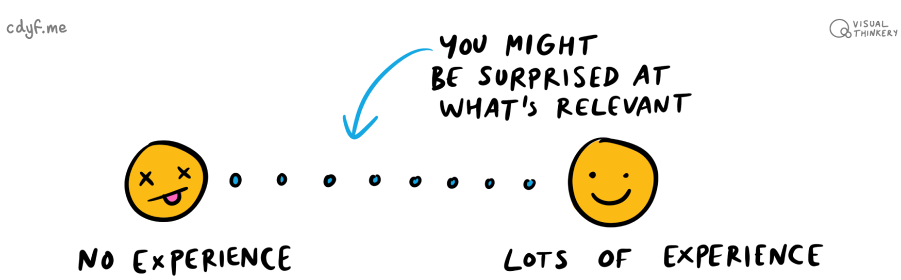


Figure 5.2: You might be surprised by which of your experiences are relevant, and what kinds of experience are relevant on your CV. What's relevant sketch by Visual Thinkery is licensed under CC-BY-ND

Figure 5.3: There is a lot more to you than your grades. Your experience tells people much more about your character, not just paid work, but any voluntary work and projects you've been involved in too. I am more than just my grades sketch by Visual Thinkery is licensed under CC-BY-ND

5.3 Are you experienced?

So what counts as experience?

- Freelance work: being self-employed
- Insight programmes and spring weeks: work shadowing
- Part-time jobs: casual or part-time work

5.3.1 Big name experience

It's easier than you might think to get a big name on your CV. For example, many large employers run insight days, vacation schemes and spring weeks. These are often aimed at first years, 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 5.4.

Figure 5.4: 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

Other ways to get a big name on your CV include joining a big name competition or event, for example:

- IBM hosts the annual Call for Code developer.ibm.com/callforcode
- Google hosts several events including:

    - Code Jam, HashCode and Kick Start codingcompetitions.withgoogle.com
    - Summer of Code summerofcode.withgoogle.com

- Facebook has hackathons, see facebook.com/hackathon and developers.facebook.com
- Microsoft hosts the Imagine Cup imaginecup.microsoft.com
- There are many others like these listed at devpost.com, Major League Hacking mlh.io and Hacker Earth hackerearth.com etc

Big names can look good on your CV, but they are not the only way to make your CV stand out.

5.3.2 Voluntary experience

Experience in the CV sense usually means paid work. However, experience in the context of *are you experienced?* (Hendrix, 1967) 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
- Fixing bugs in open source software

5.3.3 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 disregard it as non-technical or consider it "low-skilled". In the section on structuring your CV, (Black, 2019b) 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 11 to 18 I was a paperboy, except unlike the one selling newspapers in the street in figure 5.5, I delivered newspapers directly to the doors of paying customers every morning. This was not a particularly highly skilled job, but it *does* demonstrate:

Figure 5.5: Casual and part-time work tell an important story about your on on your CV. For example, from the age of 11, I was a paperboy, delivering newspapers door to door to paying customers. This demonstrates reliability and work ethic, because I did this in all weathers (sun, wind, rain, snow, hangovers etc) for seven years, man and boy! If you have casual experience, 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

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

If you have experience of working in retail, such as serving customers in a supermarket, this also demonstrates your ability to provide good customer service and work as part of a team. This is the "nice to have around" bit that Jonathan Black refers to (Black, 2019b) and is something your formal education will not typically provide any evidence of. So don't fall into the trap of discounting the value of casual or part-time labour.

5.4 Summarising your experience

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

This chapter is under construction because I'm using agile space station development methods, see figure 5.6.

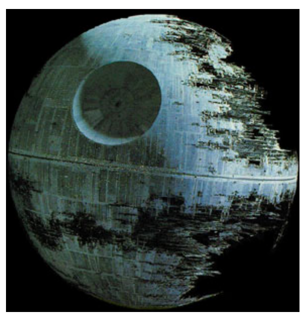


Figure 5.6: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

Chapter 6

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 computer, just 70 short years ago. 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 some of those opportunities for those with, and without, degrees in Computer Science.

6.1 What you will learn

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

- Describe why computing is a stimulating and challenging subject to study in its own right
- Identify roles where studying computer science is relevant, beyond software engineering
- Describe why NOT studying computer science doesn't necessarily "lock you out" of doing taking on some of these roles

6.2 Computing: your future?

What role will computing play in your future career? A Professor of Computer Science at Princeton University, Robert Sedgwick, has, like many others, argued that Computer Science should be required of all students:

Figure 6.1: 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

> Every college student needs a computer science course, and most need two or more. More and more educators are beginning to recognize this truth, but we are a long way from meeting the need. –Robert Sedgwick (Sedgwick, 2019)

6.3 Computational joker

If academic disciplines are playing card suits, then Computer Science is the joker in the pack shown in figure 6.2. 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. 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.

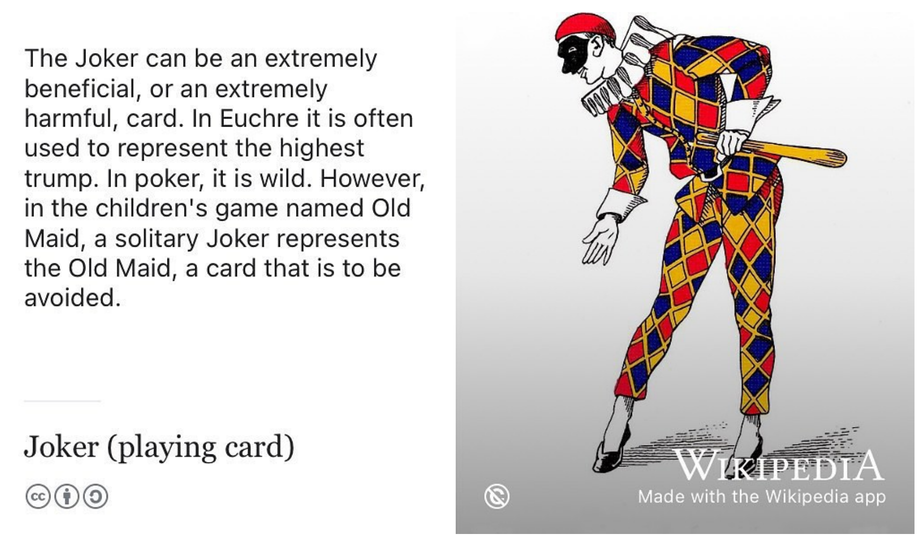


Figure 6.2: Computer Science: The joker in the pack. Public domain image of the Jolly Joker, a vintage Masenghini Italian playing card via Wikimedia Commons w.wiki/35EW adapted from the joker playing card using the Wikipedia app.

6.4 Developers in demand

Demand of software developers (jokers?) is high, on a par with teaching and nursing in terms of numbers. For example, in the UK, the most common jobs for graduates from 2017-18 are shown in Figure 6.3, based on data taken from an article on the graduate labour market in 2021 (Ball, 2020)

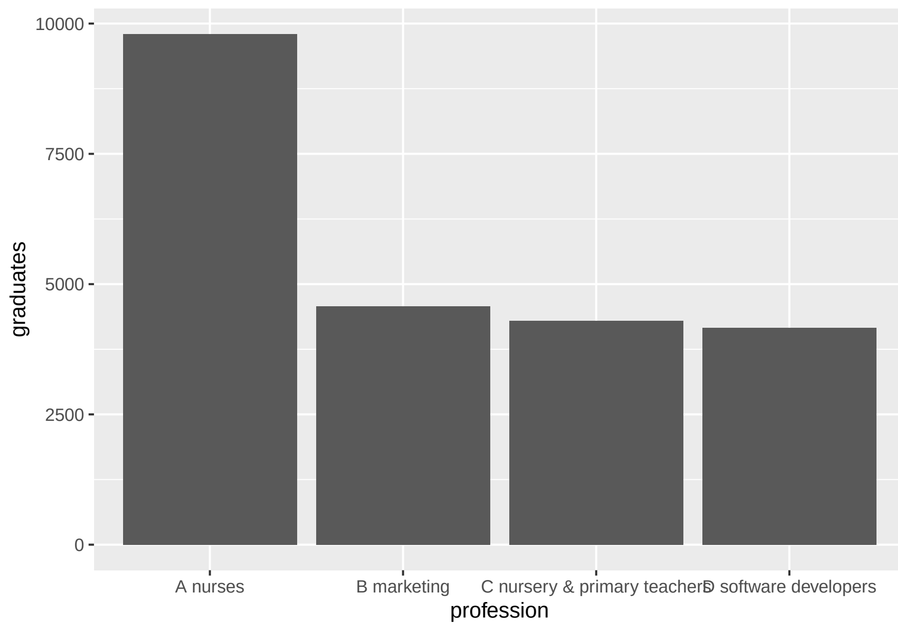


Figure 6.3: The most common jobs for graduates in the UK in 2017-18, demand for software developers is high according to data taken from (Ball, 2020)

6.5 Summarising computing your future

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

This chapter is under construction because I'm using agile space station development methods, see figure 6.4.

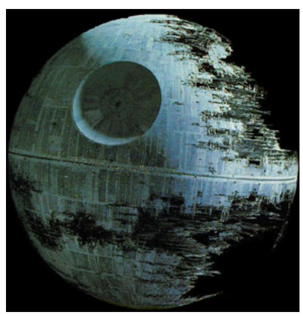


Figure 6.4: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

Part II

BUILDING YOUR FUTURE

Chapter 7

Debugging your future

It's all very well *designing* your future but now you need to actually *build* and *test* it. An obvious place to start is with your CV, because that's where most people get going. How can you create a bug-free CV, resume or completed application form? How can you support applications with a strong personal statement or covering letter? How can you debug your future?

7.1 What you will learn

By the end of this chapter you will be able to

- Structure and style the content of your CV and resume appropriately
- Describe the short stories of your relevant experience, projects and education etc
- Identify and fix bugs in CV's by:
  - Constructively criticising other people's CVs
  - Asking for, listening to, and acting on constructive criticism of your own CV
- Quantify and provide evidence for any claims you make you on your CV

7.2 Beware of the black hole

Before we get started, let's consider some advice from software engineer Gayle Laakmann McDowell shown in figure 7.2. Gayle is an experienced software engineer who has worked at the biggest technology employers in the world, Apple, Microsoft and Google. She's also authored a cracking series of books

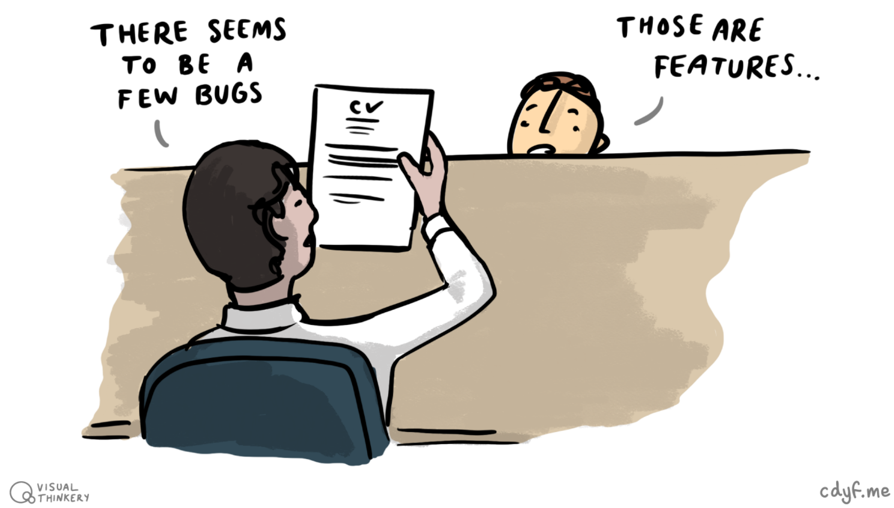


Figure 7.1: Is that a bug or a feature in your CV? To stand a chance of being invited to interview, you'll need to identify and fix any bugs in your written applications. If you don't, your application risks being sucked into a black hole and will never be seen again. Features not bugs picture by [Visual Thinkery](https://visualthinkery.com) is licensed under [CC-BY-ND](https://creativecommons.org/licenses/by-nd/4.0/)

on technology careers, particularly *Cracking the Coding Interview* (McDowell, 2015) which we'll come across in the chapter on interviewing. Gayle refers to the employer "black hole":

Figure 7.2: Beware of what software engineer Gayle Laakmaan McDowell calls the employer "Black Hole", especially if you're applying to large employers. Laakmann portrait by Gayle Laakmaan is licensed CC BY 4.0 via Wikimedia Commons w.wiki/wiu

> Getting through the doors, unfortunately, seems insurmountable. Hoards of candidates submit resumes 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 resumes that even a top candidate would struggle to stand out. –Gayle Laakman McDowell (McDowell, 2011, 2014)

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 the employment black hole. It needs to be good enough to persuade an employer to invite you to an interview. To get started, you will write this in an employer-agnostic way 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.

7.3 It's not bug, its a feature

It's an age old trope in Computer Science that engineers use to cover their mistakes, passing off their accidental bugs as deliberate features of their work:

> "It's not a bug, it's a feature!" —A. Hacker (Carr, 2018)

Nobody likes buggy software, but we unfortunately routinely 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 ~~fetal~~ 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 they are looking for reasons to REJECT your CV, rather than ACCEPT it, because that's a sensible strategy for shortlisting from a huge pool of candidates for interview. A buggy CV, application and covering letter could ruin your chances of being invited to an interview.

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

> DISCLAIMER: If you ask three people what they think of your CV, you will get three different and probably contradictory opinions. CV's can be very subjective thing. The advice below 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 reader. There are some general rules, which are described below.

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 but not give your whole life story
- Your CV will be assessed in seconds, rather than minutes so brevity really is key
- Bullet points with verbs first will:
  - allow your reader to quickly scan your CV (employers don't read CVs, they scan them) (Richards, 2019a)
  - highlight your key activities
  - avoid long sections of prose (which the reader will probably skip anyway)

7.4 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 Education
2. **STYLE**: Does it look good, decent layout, appropriate use of LaTeX or Word or whatever? Are there any spelling mistakes, typos and grammar? Don't just rely on a spellchecker, some typos can only be ~~spitted~~ spotted by a human reader
3. **LENGTH**: Does it fit comfortably on (ideally) one page (for a Résumé) or two pages (for a CV)? See length
4. **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 structuring
5. **VERBS FIRST**: Have you talked about what you have actually done using prominent verbs first in bullets, rather than just what you think you know? Avoid long sections of prose.
6. **RESULTS**: Have you also demonstrated and *quantified* the outcomes of your actions where possible, see Context, Action & Result (CAR)

7.5 Structure your CV

How you structure your CV will depend on who you are and what your story is. Recruiters at Google suggest five sections, that follow a header section. Before we look at those, lets look at some general points about CVs, watch the videos shown in Figure 7.3.

As Jonathan Black, director of the careers service at the University of Oxford has pointed out,(Black, 2019b) a key part of your story that you want to communicate in your CV is that you :

1. take responsibility
2. achieve things
3. are nice to have around

How can you demonstrate this? Watch the short video in Figure 7.4.

Figure 7.3: Recruiters at Google, Jeremy Ong and Lizi Lopez outline some tips and advice for creating your resume. (Ong and Lopez, 2019)

Figure 7.4: 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. (Black, 2019b)

7.5.1 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 7.5 for Alan Turing. That's it!

Alan Turing

alan@turing.org.uk ☎+44 (0)161 306 9280 enwp.org/Alan_Turing

Figure 7.5: Keep the header of your CV simple. Just your name, email phone number and any relevant links are all you really need. Anything else, will waste valuable space and distract the reader.

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, photo and home address aren't relevant and 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) so other contact details are irrelevant at this point. After your header I suggest you have five sections that cover the following:

1. Education: the formal stuff
2. Experience: paid work
3. Projects: side, social or University projects
4. Activities and leadership
5. Awards and honours

Let's look at each of these sections:

7.5.2 Your education

Unless you have significant amount of experience, the education section of your CV is likely to be the first real section, after the header. Your education section needs to strike a balance between:

- 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, without getting bogged down in the details.

You've invested a significant amount of time and money in getting your degree. At this stage, your degree justifies more description than a terse one line "*BSc*

Computer Science". You'll need to say more than Pen Tester and Rick Urshion (for example) but not as much as Mike Rokernel, who has given *way* too much information on his degree. You don't need to name every single module and give a mark for each. Neither do you need to give two decimal places (it happens). Pick out relevant modules, or the ones you got most out of. Employers 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.

7.5.3 Your experience

Experience is where you can talk about any paid or voluntary work experience you have. Don't discount 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, 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 7.6: Are you experienced? What have you done outside of your formal academic education? Experience sketch by Visual Thinkery is licensed under CC-BY-ND

If you don't have much experience, don't worry, there are plenty of opportunities to get some. For details, see are you experienced?

7.5.4 Your projects

The word "Projects" is a powerful catch-all description for capturing all other things you get up to. These might include:

- personal side projects
- social responsibility projects
- entrepreneurial projects
- University projects

They will most likely be unpaid because paid work fits better in your experience. 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, Dani Stefanovic's 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 7.7.

Another good source of projects is to fix a bug in an open source project. It might sound trivial, but fixing a bug demonstrates that you can collaborate with others, understand the toolchain being used (which might be complex) and solve problems.

If you've done group projects at University, be sure to mention them. Try to be *more* descriptive than this:

- "First year team project"
- "Second year team project"
- "Final year project"

Those project names are pretty opaque and don't give the reader much to go on. What was the context, action and result (CAR) of the project? 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!

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

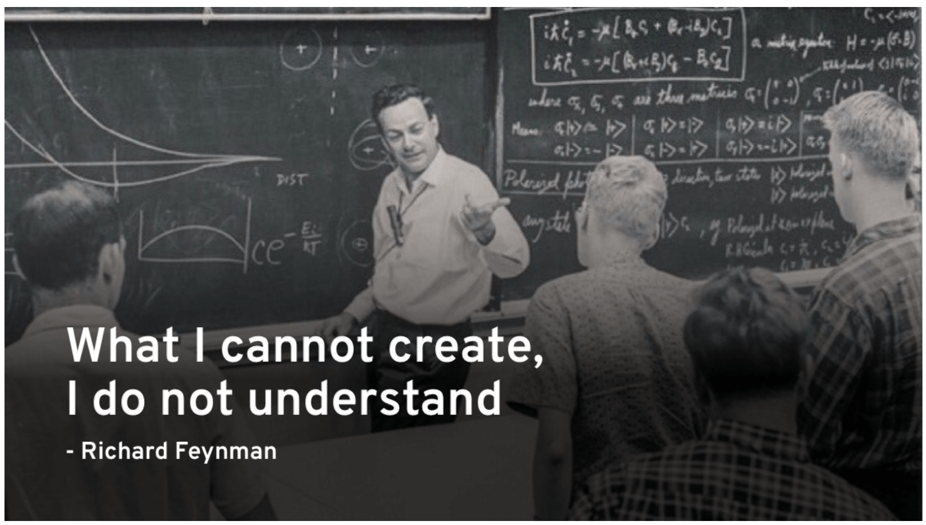


Figure 7.7: Richard Feynman once chalked "What I cannot create, I do not understand" on his blackboard at the California Institute of Technology while teaching the The Feynman Lectures on Physics. (Way, 2017) Creating software and hardware in personal side projects is a great way to build new knowledge and help your CV stand out like a sore thumb (in a good way) . Public domain image via Dani Stefanovic github.com/danistefanovic/build-your-own-x

- WidgetWasher is a web service that takes widgets and washes them before painting them a random colour
- Tested WidgetWasher on a range of different operating systems and devices
- Collaborated with one other contributor for two days at a hackathon using pair programming
- Designed and implemented an API using a RESTful approach

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

- Designed and implemented an API using a RESTful approach
- Collaborated with one other contributor for two days at a hackathon using pair programming
- Tested WidgetWasher on a range of different operating systems and devices
- WidgetWasher is a web service that takes widgets and washes them before painting them a random colour

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

7.5.5 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. Don't do it! Why not? Let's imagine, that like Rick Urshion, you include Python in a long list skills, with its own dedicated section. There are at least four problems with Rick's not so skilful approach:

1. **No context** to give the reader an idea of the 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** or *evidence* to back up his claims of having skill with Python. So Rick claims he knows Python. So what? What did he *do* with those python skills? We don't know because he hasn't told us. Perhaps he DOES have Python skills, perhaps he DOESN'T. Is he telling lies and peddling fake news? It's difficult to tell.
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 tailoring** the skill to the job that he's applying for. Python would be valuable if he was applying to be a data scientist, software engineer, or automated tester but perhaps less relevant for a business analysis, consultancy, user experience or project management roles

5. **No story** told for that skill. This makes for a very dull and boring read. Yawn. Next!

So, this doesn't mean Rick shouldn't mention his Python skills. He needs to give us the context, action and result (CAR). This will make his pythonic story much more convincing and interesting to read. Giving the CAR will improve his chances of being invited to interview.

This applies to soft skills too, not just hard technical skills. Best to mention the context in which you've used any 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 don't dedicate a whole section to it
- stick to your strongest and most relevant skills that you are comfortable to answer questions on in your interview
- Microsoft Office is *not* generally a very interesting skill because everyone has it. Don't waste space talking about it unless you've done something very advanced with it, like some complicated integration with other software.

If you're a computer scientist, you also have demonstrable "meta" skills like the ability to learn things quickly. You can also think logically, reason, problem solve, analyse, generalise, criticise, 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.

7.6 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.

7.6.1 Your style

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

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

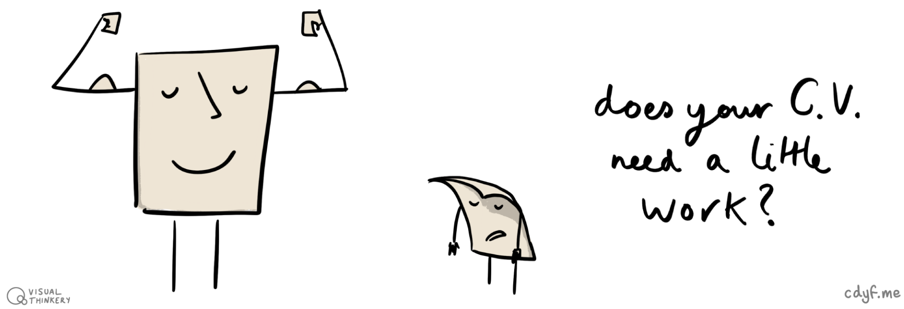


Figure 7.8: Does your CV need a little work? The truth is your CV is never finished, you will be continuously developing, debugging and releasing it throughout your life. It's such a crucial document because it will determine if you are interviewed, so its important to spend time getting it right. CV work sketch by Visual Thinkery is licensed under CC-BY-ND

mangled by transmission via the web or email. 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 any smartphone or tablet and you'll see what I mean. It helps if you can give the file a descriptive name so `ada_lovelace.pdf` is a better filename than `my_cv.pdf`

It's fine to author your CV in Microsoft Word, but you'll want to save as PDF to make it more platform independent. LaTeX and overleaf can be used to create professional PDFs and have many templates, see Getting started with LaTeX: LaTeX4year1 if you've not used LaTeX before (or you need to refresh your memory). (Hull, 2020)

7.6.2 What's your story?

One way to structure descriptions of items within each section of your CV is to use **C**ontext, **A**ction and **R**esult (CAR) sometimes called STAR (**S**ituation **T**ask **A**ction **R**esult) to tell your stories. This 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 should tell the reader (really spell it out) more about the context in which you've used python, what you actually did with it and what the result was

- **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?
- **ACTION**: What did you *do* with Python? Did you use some particular library? Did you integrate or model something?

- **RESULT**: What was the result and how can you measure it? You picked up some knew skills? You made something that was inefficient and awkward into something better, cheaper or faster?

Similar to CAR and STAR, recruiters at Google (see Figure 7.3) advise candidates to describe things using “Accomplished [X], as measured by [Y], by doing [Z]”

So instead of saying

> “generated reports for end users.” -

You could say

> “Generated daily reconciliation report for team by automating workflow of 8 different tasks”

The second is better because its more specific, captures the result (accomplishment), by quantifying it (“as measured by”) and talks about the actions (the doing part).

7.6.3 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 7.9. It is also advisable to create a one page Resume. (David, 2017)

At this stage in your career you should be able to fit everything on to one page. It can be challenging squeezing it all on:

> “If I had more time, I would have written a shorter letter.” —Anon

It takes more time to write less. Writing a one page resume 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 to have a strong one-page resume than a weaker two-page CV that is padded out with filler to make up the space, as described in the video in figure 7.9. 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 7.10.

If you’re struggling to fit all the information onto a one page resume, 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, its a good store of stuff you might want to add to customised one-page resumes.

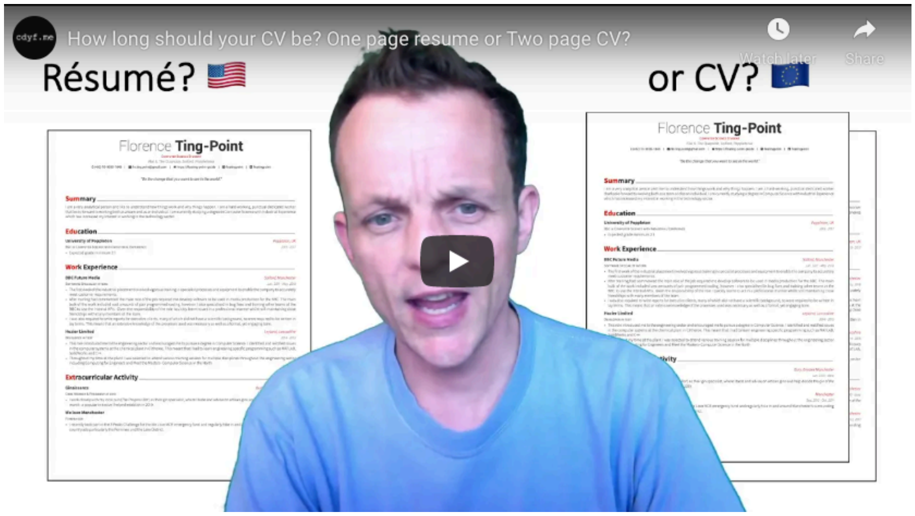


Figure 7.9: How long should your CV be? Should you write a two page European style CV or an American style resume (one pager)? [@youtube-resume-or-cv]

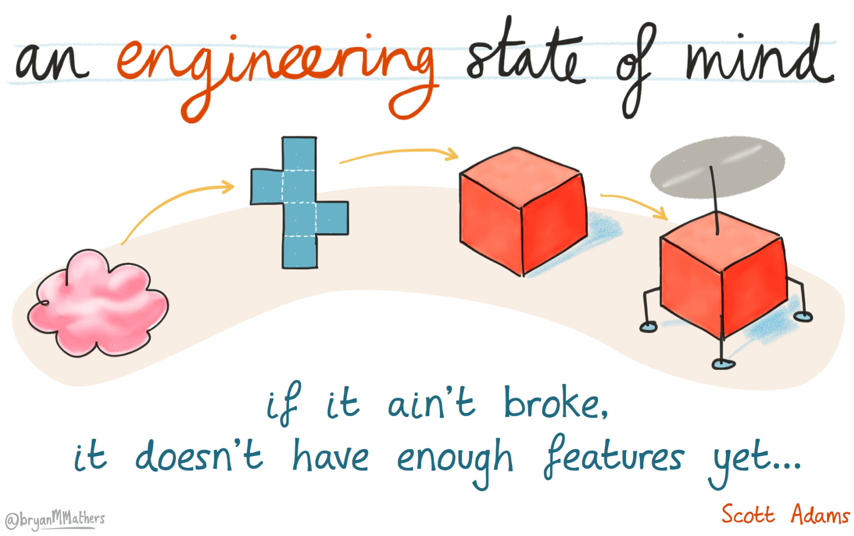


Figure 7.10: If it ain't broke it doesn't have enough features yet. Adding more features to software doesn't necessarily make it better. Likewise, adding more pages and content to your CV or resume won't always improve it. It's often better to be precise and concise, rather than bloated and potentially more buggy. An engineering state of mind by Visual Thinkery is licensed under CC-BY-ND

7.6.4 Verbs first

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 to test and build an open source Massively Multiplayer Online Role-Playing Game (MMORPG)

you could say:

> "Built and tested a large open-source codebase using Eclipse, Ant, JUnit and Jenkins …"

followed by:

> "Added and deployed new features to a Massively Multiplayer Online Role-Playing Game (MMORPG) in Java…"

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, but too many can look clumsy and give the wrong impression.

7.6.5 Your links

Augmenting your CV with web links (hyperlinks) adds context, without adding too many words or taking up valuable space as shown in figure 7.11

Links are also a great way to add evidence and substantiate any claims that you make. They allow your reader to *read between the lines* and make inferences from the information you've provided them with. For example, you might says things like:

- *"Built a thing"* reading between the lines: "I like building things. Look at this thing I built just for fun, its really cool"
- *"Joined an organisation called bla"* reading between the lines: "I was part a bigger thing you might not have heard about but you can find out about here"
- *"Competed this event called bla"* reading between the lines: "I really enjoy learning from other people by going to hackathons"

LinkedIn: https://www.linkedin.com/in/ad a-lovelace-038b37/
↓
LinkedIn: https://linkedin.com/in/ada-lovelace-038b37/
↓
linkedin.com/in/ada-lovelace-038b37
↓
linkedin.com/in/lovelace

Figure 7.11: Adding links is a good way to augment your CV. If you're adding LinkedIn, make sure you customise your public profile URL, to remove the default random alphanumeric string at the end, like the `038b37` example here. (Hoffman, 2020) You can also remove any ugly `http::`, forward slashes `//` and `www` in URLs which are distracting noise. 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 its a LinkedIn profile.

- etc

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:

- hyperlinks are readable and descriptive (Richards, 2019b)

- hyperlinks are clickable in the PDF. Don't make your reader cut and paste (or even type) URLs, they are too busy. If they are clickable, people are much more likely to follow them
- hyperlinks are paper-proof. Some people still print CVs so the phrase "click here" won't work well on printed paper. See to print or not to print — a CV, that is (Garone, 2014)

Besides LinkedIn you could include public profiles from github, stackoverflow, (Hamedy, 2019) devpost, hackerrank. You can also link to personal projects 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.

7.6.6 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 "resume robots" (if you like) are unlikely to have arms and legs like the one in Figure 7.12, but they *will* be looking for keywords and standard headings in your CV. You can get automated feedback from a range of different robotic systems, though it is a good idea to remove any personal information like phone numbers etc before using these free services. You might also want to check what the services privacy policy says about what they do with you personal data. Resume robots include:

- careerset.io, a free service provided by a UK based company, careerset Ltd.
- resume.io, a free service provided by a Dutch company, Imkey BV
- jobscan.co, a free service provided by an American company

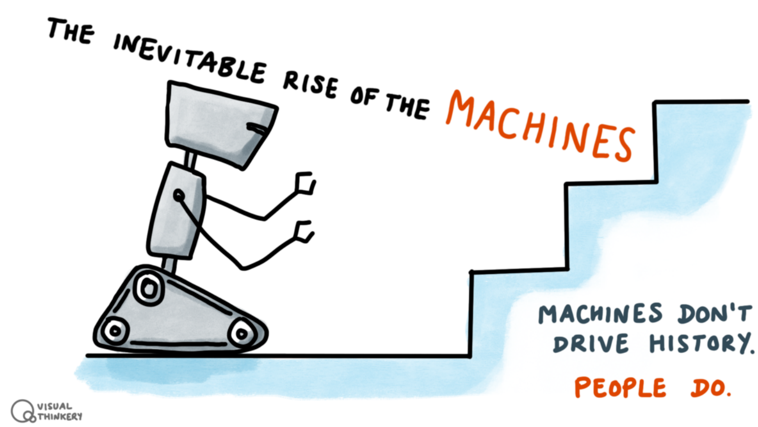


Figure 7.12: Although they often struggle to get up the stairs, resume robots are likely to play an important role in any decisions to invite you to a job interview, especially if you're applying to bigger companies. Make sure your CV is resume robot friendly by feeding it through a robot . Machines by Visual Thinkery is licensed under CC-BY-ND

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 identify it. 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?

7.6.7 Your references

You might be tempted to put your referees details on your resume. 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. They are typically taken up much later, when you've been offered or are about to be offered the job
- references give personal information out. 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.

7.7 Breakpoints

Some breakpoints for discussion:

- How long should your CV be?
- Should you have a personal statement on your CV?
- One column or two column layout?
- Should I put education or experience first?
- How many of my hobbies and personal interests should I list?
- How many employers actually read cover letters?

7.8 Big Bad Buglist

A quick check-list of commons bugs

1. Has it been reviewed by several people?
2. Have you reviewed other people's CV's?
3. Does the style look good?
4. Is there too much or too little whitespace?
5. Is there anything irrelevant on there? e.g. personal information, swimming certificates from ten years ago
6. Is your year of graduation, degree program, University and expected (or achieved) degree classification clear?
7. Are there any spelling mistakes or grammatical errors?

8. Is it in reverse chronological order?
9. Have you used too many personal pronouns?
10. Have you made it clear what you have actually done using prominent verbs?
11. Have you mentioned courses you are studying now (or next semester)?
12. Is it targeted to an employer or job description?
13. Is it balanced, with all non-technical
14. Does it have a good, clear structure? Typically education, experience, projects etc
15. Have you distinguished between paid and unpaid or voluntary work?

7.9 Covering letters & personal statements

Applications and CV's are often accompanied by covering letters or include some kind of personal statement. Whereas a lot of your CV is essentially a bulleted list of facts and statements, a covering letter or personal statement gives you a chance to *really* demonstrate your fluent written communication skills in clear prose. Let's say you're applying for a widget engineering position at BigWidget Inc. There are three things you need to cover in approximately this order

1. **Why them?** Why are you applying to BigWidget Inc.
2. **Why that role?** BigWidget Inc. employees have many roles and responsibilities, so what is it about widget engineering that attracts you?
3. **Why you?** Why should they employ you? What makes you stand out from all the other widget engineering candidates? What is your Unique Selling Point (USP) or points?

7.9.1 Does anyone actually READ covering letters?

Some employers will read them very carefully, some less so. It's not clear which employers will bother and which won't.

Even if nobody reads your covering letter, it is still worth writing one because it forces you to rehearse standard interview questions shown above. See e.g. www.careers.manchester.ac.uk/applicationsinterviews/cl/ for further information. Think of it as practicing your elevator pitch.

7.10 Debugging summary

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

This chapter we've looked at how to debug your CV. It's important to try and squash any of the bugs we've described here, before an employer see's your CV.

Chapter 8

Finding your future

So you've successfully debugged your future. 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?

8.1 What you will learn

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

- Formulate job search strategies, by role, by sector and location
- Describe the timing of recruitment cycles used by employers
- 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
- Apply your search strategies to advertised (and unadvertised) opportunities

8.2 Where can you for look for jobs?

The marketplace for job searching and job hunting advice is incredibly crowded. Employers spend huge amounts of money recruiting staff and this is reflected in the enormous range of job websites, which are often accompanied by advice on job hunting. I've broken resources down here into three categories, student specific, more general resources and recruiters.

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

8.2.1 Student and graduate specific resources

These resources are specifically aimed at undergraduate students and graduates:

- 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
- 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
- varsitycareershub.co.uk, targeting students from Loxbridge but many of the employers recruit much more widely
- Year in Industry if you're looking for a year in industry
- Your University careers service, e.g. if you're studying at the University of Manchester it's careerconnect.manchester.ac.uk. University jobs boards are good places to look for opportunities that are specifically targeted at the University where you are studying

8.2.2 More general resources

These resources are aimed at a wider audience (not just students and graduates) but are useful nonetheless.

- Google job search aggregates content from many (but not all) of the sources listed above. 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 based on schema.org/JobPosting. Keywords like `job`, `intern` and `placement` in an regular google search will trigger the job search product:

  -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+placement

-If you're looking for a job AT google, they have moved from jobs.google.com to careers.google.com

- LinkedIn advertises job vacancies, is frequently visited by recruiters and you can often apply for jobs directly on LinkedIn (athough making fast applications is not always a good thing). See university.linkedin.com/linkedin-for-students and linkedin.com/learning/learning-linkedin-for-students

- 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.

- stackoverflow.com/jobs jobs via StackOverflow. You've cut and pasted the code… they advertise technical vacancies too

- Otta.com for people with 0-10 years experience. From engineering to sales, discover jobs & internships at London's most innovative companies.

8.2.3 Recruiters

Recruiters can help you find work and they operate in every industry sector. There are two basic kinds of recruiters 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.

Recruiters are not always technical people, so don't expect them to have lots of knowledge about 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 8.2, especially if they work for an agency rather than being directly employed by the organisation you are interested in.

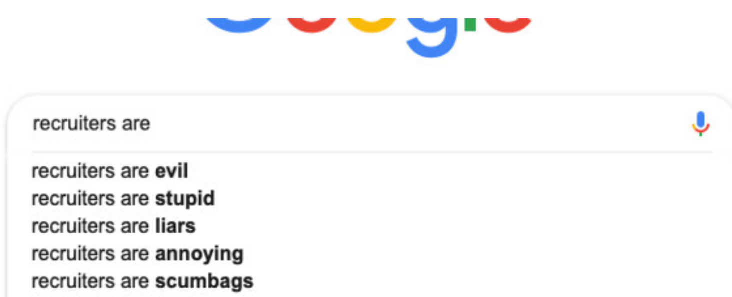


Figure 8.2: The autocomplete algorithm of a well known search engine gives you an idea of what *some* people think about *some* recruiters. This doesn't mean you should avoid recruiters, just be careful how you use them.

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, but it is worth remembering that some recruiters work primarily for their clients (employers) not you.

8.3 Job search strategies

It is a good idea to have a range of different strategies for job hunting, and change the strategy during the year. Before we look at strategies, it's important to realise that when you're reading job descriptions for vacancies, some employers will over state 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 truth is, if you've got 75% of what they are asking for, you should definitely apply.

8.3.1 Over specified job descriptions

Employers and recruiters routinely over-specify job descriptions. A good example of this is, when the Swift programming language was 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 are going to have *long* search trying to find their unicorn. It will probably take them at least five years!

Table 8.1: Where, when and how to apply for vacancies, internships, placements, graduate jobs and schemes in large multinational corporations and small to medium sized enterprises (SMEs)

	Large corporations	SMEs
Where	Advertise broadly	Less likely to advertise on big jobs bo
When	Vacancies earlier in the academic year	Vacancies tend to be later in the acad
Type	Unlikely to consider speculative applications	May consider speculative or informal
How	Multistage applications, several rounds of interviews	Shorter application processes

> Job requirements: pic.twitter.com/4QB1oPyxU0
>
> — Mubeen (@Mubeeen_) July 28, 2020

The moral of the Apple story above is, if you don't meet all the criteria on a job spec, that shouldn't stop you applying. Most job specifications are over-specified as wishful employers dream up their ideal candidate.

At the beginning of the academic year in September you might target large multinational organisations. If you're not successful, you can switch to smaller employers later in the academic year. Table 8.1 shows summarises some of the when and where some employers advertise.

8.4 Breakpoints

Take a pause and check the values of your parameters and variables. Here are some breakpoints for reflection and discussion:

- How many jobs should you apply for?

- 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?

8.5 The power of weak ties

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)

Figure 8.3: It’s not what you know, its who you know. 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 create opportunities for that to happen. Bumped into sketch by Visual Thinkery is licensed under CC-BY-ND

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 8.3. “Bumping into” here, could mean either physical or virtual.

8.6 Summarising search

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

We’ve looking at search techniques that will help you find opportunities you care about.

This chapter is under construction because I’m using agile space station development methods, see figure 8.5.

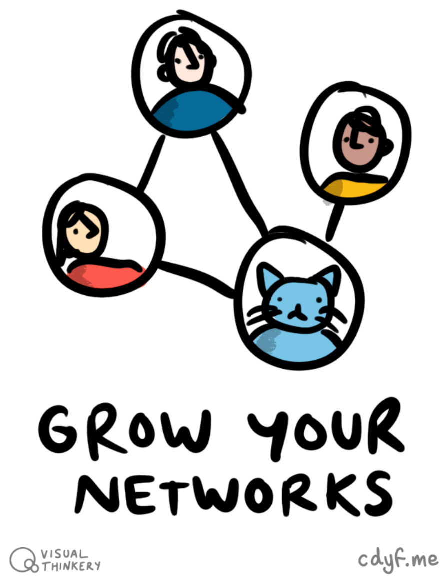


Figure 8.4: Who is in your network? Grow and use your network, both the strong ties and the weak ties. Weak ties are often the most important when it comes to job hunting. Networks sketch by Visual Thinkery is licensed under CC-BY-ND

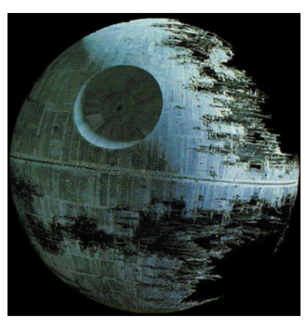


Figure 8.5: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

Chapter 9

Speaking your future

Congratulations, you've been invited to an interview. It might be a telephone, video or face-to-face, or it might even be 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 CV, along with any accompanying covering letters, application forms or digital portfolios, have hit the target. All that reading and writing has paid off. BULLSEYE!

Having passed the first stage, you move onto speaking and listening which you'll do in a live interview. One of your goals is to convince the interviewers that you can articulate yourself clearly, and communicate well using spoken natural language, while listening carefully to questions that they ask. Remember this is part of fundamental skill we talked about in communication I/O

If you've got an interview, you can feel good about having a bug-free CV shown in Figure 9.1. But now you have a new set of problems. 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 anyway and are they the kind of people you actually want to work with everyday? You'll 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
- something like the next two years of your life (Black, 2017) as the first part of up to 80,000 hours

So you want to ensure employers are a good match and not going to waste your time.

Figure 9.1: 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

9.1 What you will learn

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

- Identify kinds of interviews you might be invited to
- Anticipate common interview questions, technical and non-technical
- Prepare questions for your interviewer by researching the employer
- Formulate strategies for negotiating job offers
- Calm your interview nerves

9.2 Interviews

Broadly speaking there are two basic kinds of interviews

- technical or coding interview
- non-technical interview, sometimes called competency based interview or HR interview (human resources)

These can be conducted in various modes:

9.2.1 Modes of interview

Interviews can be conducted in various modes:

- telephone (no visual contact)
- pre-recorded pieces to camera (this is you talking to your webcam)
- teleconference (zoom / teams etc) with cameras and microphones turned on
- real-time face to face

9.2.2 Competency interviews

Competency interviews are there to test your soft skills, find out what you're like, how you work in a team, if you can communicate well. 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 can you play in a team?
2. Tell me about a time when you showed integrity and professionalism

Figure 9.2: Remember to do the mute / unmute toggle dance if you have a teleconference interview on zoom or similar. Bob... by Visual Thinkery is licensed under CC-BY-ND

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 handled conflict in the workplace
6. How do you maintain healthy working relationships with your colleagues?
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.

For more exam * google.com/search?q=competency+based+interview+questions * careers.manchester.ac.uk/applicationsinterviews/interviews

Since there's already tonnes of information The rest of this chapter will focus on technical interviews, also known as coding interviews.

9.2.3 Coding interviews

There are a lots of resources to help you prepare for and practice coding interview questions, the best 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, University of Manchester Computer Science graduate Petia Davidova explains in figure 9.3 what she learned from failing several coding interviews at big technology companies.

Figure 9.3: Petia describes her worst software engineering interview failures. (Davidova, 2021) Petia demonstrates her growth mindset and productive failure(s). Although she failed her interviews, she learned lots from the process and went on to get a job she wanted.

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 things will help both your general professional development and your chances of success in future technical interviews.

9.2.4 Project Euler

Project Euler provides a wide range of challenges in computer science and mathematics. The challenges typically involve solving a mathematical formula or equation, see projecteuler.net

9.2.5 Hacker Rank

Hacker rank 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.

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

9.2.6 Topcoder

TopCoder is a platform for competitive programming online. You can complete on your own directly online using their code editor. Single Round Matches are offered a few times per month at a specific time where you compete against others to solve challenges against the clock, see topcoder.com

9.2.7 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

9.2.8 Leetcode

LeetCode is a platform to help you enhance your skills, expand your knowledge and prepare for technical interview see leetcode.com.

9.2.9 Pramp

Pramp offers free mock technical interviewing platform for engineers. Pramp, **Pra**ctice **m**akes **p**erfect, 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

9.2.10 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. In it's own words the 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)

ICPC is organised by the Association for Computing Machinery (ACM), a global organisation 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.

9.3 Breakpoints

You should find out as much as you can about your potential employer before the interview. Here are some breakpoints, pause your program and check these parameters and variables:

- What the main products and services that the organisation provides?
- Who are their clients or customers?
- Who are their biggest competitors?
- 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?

9.4 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 space station development methods, see figure 9.4.

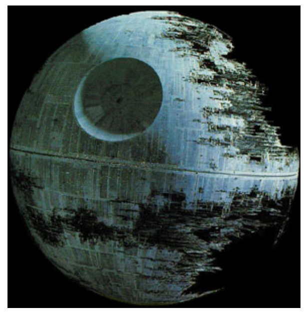


Figure 9.4: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

Chapter 10

Broadening your future

Do you feel like the *weird edge case* pictured in Figure 10.1? 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 want to:

- use your technical skills responsibly and ethically to make the world a better place?
- start your own business and make money for yourself, rather than other people?

This chapter on your alternatives will broaden your 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 their own business needs. The people who build are often employees that given organisation, 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)

10.1 What you will learn

- Describe the less obvious careers that computer science can lead to, besides software engineering

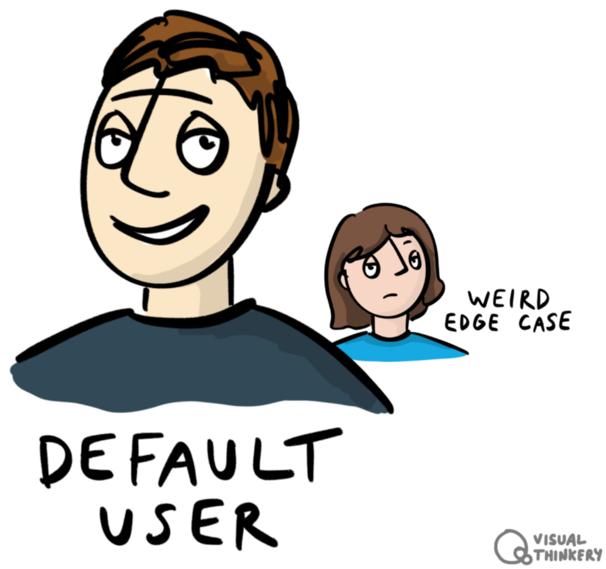


Figure 10.1: Are you a weird edge case? By default, many graduates choose a graduate scheme with big brand, often a blue-chip multinational employer. While working for these kind of employers has many benefits, they are not the whole story. This chapter looks at some of the alternatives. Default user by Visual Thinkery is licensed under CC-BY-ND

10.2 With great code comes great responsibility

Computer scientists wield tremendous power in the twenty first century:

- With great power comes great responsibility (Parker, 1962)
- With great code comes great responsibility (Goldman and Schlesinger, 2018)

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?

Figure 10.2: 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 dialog box sketch by Visual Thinkery is licensed under CC-BY-ND

10.3 Summarising your alternatives

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

This chapter is under construction because I'm using agile space station development methods, see figure 10.3.

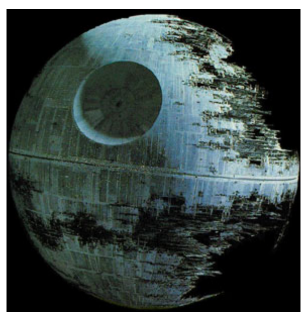


Figure 10.3: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

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 world of employment. This *might* be your first SERIOUS job, so what do you need to survive? Even better, how can you thrive in your new role and take on the challenges that are coming your way? How will you optimise your trajectory to reach new heights?

11.1 What you will learn

At the end of this chapter you will be able to

- Survive in a workplace environment
- Thrive in a workplace environment
- Avoid diving in a workplace environment
- Manage your manager

11.2 Survive, thrive or dive?

Starting a new job is a bit like starting a new relationship. You've been through the courtship of recruitment, this might have been quick or may have had many rounds of first and second dates. Now you're both committed to each other and starting the relationship for real. Simply put, there are three scenarios for you as a new employee. You'll survive, thrive or dive.

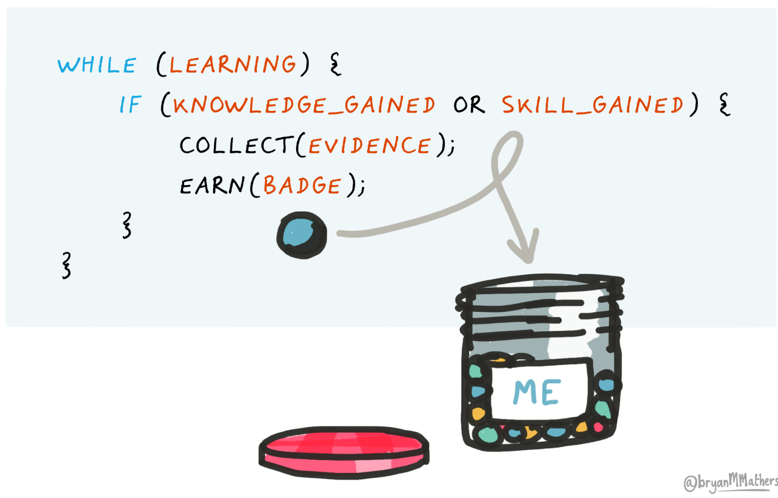


Figure 11.1: Your education doesn't finish when you start work, it should be a loop where you constantly acquire knowledge and skills during your career. You don't stop learning when you leave University. Computing Badges by Visual Thinkery is licenced under CC-BY-ND

11.2.1 Survive

It will go OK, you'll meet the expectations of your employer and become and valued employee. Most employees fit in to this category.

11.2.2 Thrive

It 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

It will go badly, you will struggle to fit in and won't meet the expectations of your employer. Once you were star-crossed lovers, but the relationship has turned sour and may end in 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. (UK, 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. (UK, 2020b)
- **It's not me, it's you**: If things get really bad, your employer may take disciplinary action against you (UK, 2021a) and in the worst case scenario, you'll be fired (dismissed). (UK, 2021b)

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 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.

11.3 Placement visits

During your placement you will be visited by an academic member of staff who will meet with you and your manager at the same time.[^If you want to meet with your academic tutor without your manager being around, please let your tutor know] The visit is a good opportunity to reflect on what progress you have made during the year alongside areas you need to improve. You will probably already have had similar meetings with your manager or managers, for examples as part of your personal development review (or whatever your employer calls performance reviews). There are three basic questions your tutor will ask you:

1. What have you been doing?
2. WWW: What Went Well?
3. EBI: Even Better If?

11.3.1 What have you been doing?

Briefly describe your roles and responsibilities during your placement to date. What projects have you worked on? What were the main technologies that you used?

11.3.2 WWW: What went well?

Are there any projects you are particularly proud of? What new knowledge or skills have your learnt 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.

11.3.3 EBI: Even Better If?

What areas have you identified for improvement in the future? Again, this includes non-technical as well as technical skills.

11.3.4 Managers comments

Your tutor will ask your manager to ask the same questions. Are there any differences between your view and your managers view of your placement? If so, why do you differ?

11.3.5 Placement report

Both your tutor and you need to fill in the placement report, no more than two pages. See the industrial placement report template on overleaf

11.4 Summarising survival

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

This chapter is under construction because I'm using agile space station development methods, see figure 11.2.

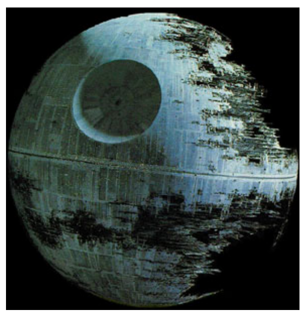


Figure 11.2: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

Chapter 12

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 are they 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 the next Ada Lovelace or Alan Turing?

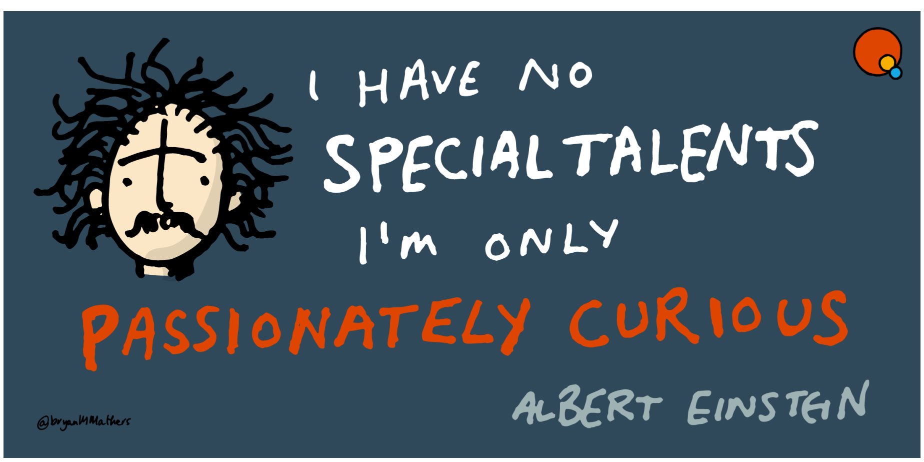


Figure 12.1: Are you passionately curious? Is further study or research the right path for you? Are you the next Einstein or Einsteiness? This chapter looks at the options. Curiosity by Visual Thinkery is licensed under CC-BY-ND

12.1 What you will learn

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

- Describe the costs of postgraduate study and research
- Describe the benefits of postgraduate study and research

12.2 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 also lists PhD opportunities, not just in the UK

12.3 Breakpoints

Some breakpoints for discussion:

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 improve career prospects?
4. Is postgraduate study and research really worth all the pain and suffering?
5. What careers can a PhD lead to?

12.4 Signposts from here on research

A good place to start if you're thinking about doing a PhD is *How to get your PhD: A Handbook for the Journey* by Gavin Brown. (Brown, 2021) 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.

> "How to get Your PhD: A Handbook for the Journey" (@OUPAcademic https://t.co/Q9dTmWQ9iF) is out today. Preview on Google Books https://t.co/US8O2EeaQf / #howtogetyourphd (14/14)
>
> — Gavin (@profgavinbrown) March 1, 2021

12.5 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 space station development methods, see figure 12.2.

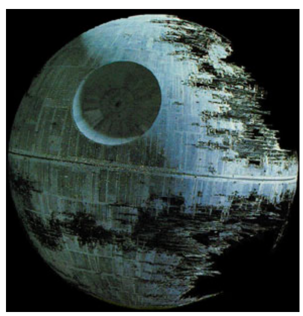


Figure 12.2: This chapter is under construction. Image of the Death Star via Wikimedia Commons w.wiki/32PB

Part III

SUPPORTING YOUR FUTURE

Chapter 13

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 a series of *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) for this guidebook.

13.1 Know who you are

There is a lot more to you than your degree. Yes, you've spent three or four years getting your degree. identify your weaknesses and work out how to improve them.

13.2 Look after yourself

Look after yourself mentally and physically.

Choose your reference points carefully, try not to compare yourself to the person at the top of the class. Ask yourself, am I doing better than last time?

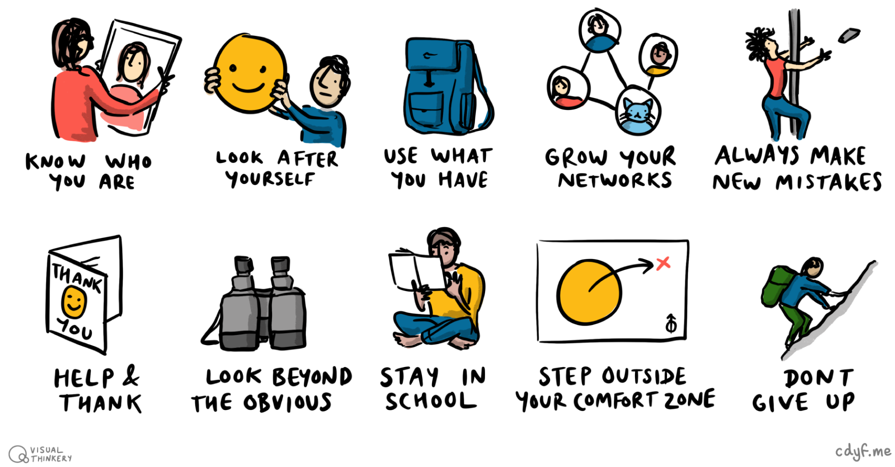


Figure 13.1: Ten Simple rules for coding your future. Know who you are, look after yourself, use what you have, grow your networks, always make new mistakes, help and thank, look beyond the obvious, stay in school, step outside your comfort zone and (*most importantly*) don't give up! Figure by Visual Thinkery is licensed under CC-BY-ND

13.3 Use what you have

This rule is borrowed from software engineer Greg Wilson @gvwilson, who probably adapted it from a quote frequently misattributed to President Theodore Roosevelt (Brewton, 2014). The full quote is

> Start where you are, use what you have, help who you can, see https://third-bit.com

13.4 Build your network

Build your network, use all the people you can. Remember that the weaker ties in your network may be more important than your stronger ties, especially when it comes to finding jobs.

It's not what you know, its who you know.

13.5 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 that 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 conciously 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 conciously reflect on why you made them. This is part of the growth mindset we discussed in the chapter on your well-being.

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 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 (top grades) rather than progress and learning. (Lahey, 2016) So as Ester Dyson says,

Always make new mistakes

- 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

13.6 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.

Join a team by helping someone, be a team player, help others, thank others for their help.

13.7 Look beyond the obvious

Be flexible in approach. Not just big employers, there are startups, SME's you've never heard of. It's not just London, and other big cities. Look beyond graduate schemes, look beyond graduate jobs.

You are not just a techie, Either.

13.8 Stay in school

Never stop learning.

13.9 Step outside your comfort zone

Am I being insensitive asking people to step outside their comfort zone when we've all be stretched beyond breaking point during COVID-19?

We're going to need to continue to step outside of our respective comfort zones in order to meet the challenges we face around the world.

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.

13.10 Don't give up

Learn to live with rejection, don't take it personally

13.11 Ten simple summaries

This chapter is under construction, but hopefully you can get the general gist of what's going on.

Chapter 14

Reading their future

It's very easy to overlook mistakes in your own writing. That's true of any written communication such as a covering letter, personal statement, email or message that you write. Overlooked mistakes are particular common in CVs (or resumes) because you can spend *hours* carefully honing the words and the formatting but not see a fatal error at the top of page one.

14.1 Debug their CV

There's at leat three solutions to this problem,

1. dogfooding we've already talked about, read your work ALOUD
2. "I'll show you mine if you show me yours."
    - Debug a friends or peers CV by swapping with them.
3. Debug the anonymised CVs below

So, here are some anonymised CVs for you to debug, from students of Computer Science. They are based on CVs I've seen, warts and all, with personal information removed. Can you spot their triumphs and tragedies? Can you debug their CV against the sample job description at CoolTech below? Can you find their future?

Special thanks to Toby Howard and Sean Bechhofer for coming up with some of these names. Please direct any complaints about the terrible geeky puns to Toby and Sean! 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.

14.2 Breakpoints

Some breakpoints for discussion, 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 improved
3. **Their Rank** Who is top of your list to interview? Who is going in the bin and why?

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?

14.3 Sample CVs

14.3.1 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 hates that horribly overloaded word.

See cdyf.me/Penelope_Tester.pdf

14.3.2 Rick Urshion

Rick is a big fan of functional programming and loves expressing himself using languages like Lisp, Haskell, Clojure, Erlang and Scala. He really hates 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 cdyf.me/Rick_Urshion.pdf

14.3.3 Marge Conflict

Marge loves version control and her superpower is resolving people's differences.

(Marge is currently working on her CV)

14.3.4 Michael Rokernel

Mike loves operating systems, but not if they get too bloated.

See cdyf.me/Mike_Rokernel.pdf

14.3.5 Florence Ting-Point

Flo loves maths and is a particularly big fan of floating-point arithmetic.

See cdyf.me/Flo_Ting-Point.pdf

14.3.6 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, Mega, Giga, Terry, Exa, Zita and Yotta. They are wildly ambitious, but critics say the Byte family have been terribly over-hyped.

See www.cdyf.me/Peter_Byte.pdf

14.3.7 Polina Morphism

Polly loves object-oriented programming. She has lots of siblings, and a cousin called, Isa.

(Polina is currently working on her CV)

14.3.8 Neil Pointer

Neil is a mature student who loves the C programming language. The Pointer family are sometimes misunderstood, but Neil compensates for this with his excellent memory management skills and efficiency. He has a younger half-brother, Neil Pointer-Exception, from his fathers second marriage. Neil Pointer-Exception prefers Java

See www.cdyf.me/Neil_Pointer.pdf

14.3.9 Bryn Hanby-Roberts

The last CV is a real one. Bryn kindly gave his permission to share it with you. He graduated in 2016, his CV is longer as he has five years of experience under his belt but it provides a useful counterpoint to the examples above.

See bryn.co.uk/cv.pdf

Thanks Bryn.

14.4 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 15

Moving your future

Finding a job often involves moving to another city or even country.

So a simple way to improve your job prospects is to broaden and deepen your job search and consider all the alternatives, geographically and otherwise.

For example, if you're seeking work or further study in the United Kingdom, there are lots of possibilities in Loxbridge (**Lo**ndon, **Ox**ford and Cam**bridge**). While Loxbridge undoubtedly offers many fantastic opportunities for professional growth and development, it does not represent *all* of the best opportunities that exist in the UK.

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. 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.

15.1 Hit the North, Not Just London

You don't have to move to London to find top employers, there are plenty located here in the North West if you'd like to stick around after you graduate and Hit the North. (Smith et al., 1987) 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

Figure 15.1: 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. Not just London Image by Sharon Dale (Dale, 2018)

employers based in the North West technw.uk if you want to find out more. To quote sharon dale, its NotJustLondon (Dale, 2018)

The list of employers in this page is currently being migrated from git.io/manc, visit that page for a list of employers from the North West.

15.2 techUK and technation.io

In addition to these techuk.org and technation.io provide more information on technology businesses outside of London.

15.3 Guest lectures from employers

The Department of Computer Science welcomes external speakers and hosts a number of guest lectures from a wide range of collaborators in industry and academia. There are several ways the department can host guest lectures

- Scheduled lectures for COMP101, a first year course with 400+ enrolled students. During 2021, these talks take place on Mondays from midday to 12.50pm on zoom
- School research seminars , see examples of past seminars
- Technical talks arranged as part of a scheduled course, speak to the course leader for the relevant course. These tend to be more advanced courses aimed at students later in their degrees or as part of postgraduate study.

- One-off talks arranged *ad hoc* that are not part of scheduled series of lectures
- Seminars and talks from speakers arranged and booked by students, for example unicsmcr.com, The University of Manchester Computer Science society

15.3.1 Examples of COMP101 lectures

Some example guest lectures for COMP101 are shown below to give a flavour of the kind of talks that are appropriate

- Hacking the Hacks, delivered by NCC Group
- Debunking the myths associated with User Experience, delivered by American Express
- The Business of Intellectual Property, delivered by Imagination Technology)
- How to Break a Hacker's Mind, Web Application Vulnerabilities Exposed, delivered by Morgan Stanley
- 100 billion ARM chips, delivered by ARM
- Software at Airbus
- Computing in the Community, delivered by CodeClub & Manchester Girl Geeks
- How to be a brilliant software engineer, delivered by Apadmi

15.3.2 Interested in speaking?

For COMP101, we are always looking for good speakers who can engage large groups of students on interesting topics that they care about and relate to Computer Science. For COMP101, there are a limited number of guest lecture slots (around 20) which run through term-time starting in November and finishing in early June. Its important that speakers

- Give much more than a sales pitch for an organisation, by providing insight into a technical subject
- Talk about content that relates to the (very broad) syllabus of COMP101
- Engage, interact, educate and entertain. Students vote with their feet (by not turning up) if they think a lecture won't be interesting
- If you would like to propose a guest lecture for COMP101, please contact Duncan Hull. For all other external seminars and events, see the links above.

Since lockdown, lectures have been online, which has allowed for more interaction, typically via the chat dialogue. We can monitor and moderate the chat, feeding questions to the speaker when it's appropriate.

Chapter 16

Hearing your future

This podcast features interviews with student as they come to end of their undergraduate degree or shortly after they graduate. We interview students to find out more about their journey from student to professional. During the interview, we ask students to tell us:

- What's their story? Tell us about themselves
- Where does their interest in computing come from
- Which organisation they were employed by, why and how did they chose them
- Tell us about their role in the organisation
- How did they find the job and what other jobs did they look for?
- What was the most enjoyable or rewarding part of working for their employer
- What advice would they give to students looking for placements
- What are their plans for the future
- What's the most interesting thing happening in computer science / technology right now?
- What are their favourite work related radio shows or podcasts?

16.1 Episode 1: Raluca Cruceru

Interview with Raluca Cruceru careers.cern/Raluca, a software engineer at CERN, will be published here shortly. Raluca graduated with BSc in Human Computer Interaction with Industrial Experience in 2020.

16.2 Episode 2: George Dunning

Upcoming interview with George Dunning, software engineer at steama.co in Manchester will be published here. George graduated with a BSc in Computer Science with Industrial Experience in 2020.

16.3 Episode 3: It could be you!

If you'd like to be interviewed, get in touch.

Chapter 17

Actioning your future

Employers are often more interested in what you have done, rather than what you know. A good technique for emphasising your actions is to lead the description of it with carefully chosen verbs. This is useful technique in CVs and resumes which will highlight what you *actually done* rather than what you *know*. See the verbs first section of the debug your future chapter.

17.1 What you will learn

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

- Emphasise your *actions* when describing your education, projects and experience
- Reflect on what skills you have, and what skills you need to develop

17.2 Breakpoints

Before you start, here are some breakpoints for you to pause and consider. Quickly scan your CV, covering letter or application form for the verbs listed in this chapter:

- Are there any verbs that are missing?
- Have you over-used certain verbs (like *worked* or *assisted* for example)
- 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 or influencing?
- What activities or projects could you do that would help you develop these missing skills?

17.3 Team verbs

Some verbs to demonstrate how you have worked and communicated with others in a team.

- administered
- advised
- assisted
- coached
- encouraged
- instructed
- interviewed
- organised
- participated
- attended
- presented
- recommended
- recruited
- suggested
- volunteered

17.4 Engineering verbs

Verbs to demonstrate your engineering and technical skills.

- added (e.g. new features)
- analysed (e.g. the requirements)
- architected
- assigned (e.g. bugs to team members)
- automated (e.g. builds and tests)
- 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)
- merged (e.g. git)
- migrated
- modified (e.g. game mods)
- specified
- tested

17.5 Leadership verbs

Some verbs to demonstrate how you have used your initiative and taken the lead:

- established
- created
- decided
- devised
- directed
- facilitated
- introduced
- launched
- led
- managed
- mentored
- motivated
- supervised

17.6 Improving verbs

Verbs that demonstrate how you have improved a situation by taking resposibility for something:

- delivered
- edited
- enhanced
- generated
- increased

- refined
- resolved
- saved
- transformed

17.7 Scientific verbs

Verbs that demonstrate your analytical and scientific skills

- assessed
- calculated
- discovered
- estimated
- evaluated
- identified
- interpreted
- investigated
- proved
- researched

- reviewed
- tested

17.8 Winning verbs

Verbs for demonstrating your achievements and honours

- achieved
- attained
- awarded
- nominated
- recommended
- selected
- mastered
- won

17.9 Planning and organisation verbs

Verbs to demonstrate your planning and organisational skills:

- arranged

- prepared
- scheduled
- organised
- planned
- produced
- revised

17.10 Influential verbs

Verbs that demonstrate how you have influenced and persuaded others:

- guided
- influenced
- liaised
- negotiated
- mediated
- promoted
- presented
- publicised

17.11 Summarising your actions

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.

Chapter 18

Scheduling your future

This is the schedule for the live (synchronous) sessions for Coding Your Future in 2021 (COMP2CARS), on **Mondays at 1pm on Zoom** where they complement the second year tutorials (COMP2TUT) at the University of Manchester.

As with semester 1, COMP2CARS takes place in the same slot as COMP2TUT when you meet your personal tutor. See the timetable studentnet.cs.manchester.ac.uk/ugt/timetable.

These sessions will revisit themes from semester one if you missed them last semester, we'll also have an open session where we'll tackle whatever issues you need about finding placements and internships during 2021. We'll be meeting at zoom.us/my/duncanhull, please sign into zoom with your @student.manchester.ac.uk account and if you can, turn your camera on.

18.1 Semester 2 dates

Semester 2 dates for 2021 are:

18.1.1 Monday 26th April at 1pm on Zoom

- *Still looking for a placement starting in 2021?* Live lecture with Q&A see also cdyf.me/finding.html

18.1.2 Mon 3rd May

Early May Bank holiday see gov.uk/bank-holidays

18.1.3 Mon 10th May at 1pm on Zoom

- *So you're going on placement in 2021?* Live lecture with Q&A, see also cdyf.me/surviving.html

18.1.4 Week 13: Monday 17th May

- Semester 2 exams: 19 May–9 June 2021
- Semester 2 ends 11 June 2021
- See manchester.ac.uk/discover/key-dates/

18.2 Cameras on or off?

I would normally expect participants in small meetings (not large ones like lectures) to turn their cameras on but I understand that there are good reasons why people may not be willing/able to and won't explicitly ask you to.

18.2.1 Why turn cameras on?

There has always been a question around whether to turn cameras on during online meetings but it is even more obvious with online meetings becoming the norm rather than the exception. I see a direct benefit in using cameras in small, personal meetings where many of us make use of visual cues to aid the flow of conversation – at the very least it's easier to identify who is talking. Additionally, it can help people get along – people might feel more 'listened to' if they can see somebody listening and personally I find it easier to remember names etc if I have a face to match the names to. I will generally have my camera on and will continue to keep it on even if I am the only one.

18.2.2 Why not?

There are lots of reasons why not. Most obviously, if you don't have access to a camera. But you may also be in an environment which you prefer others not to see, you may have anxiety around the issue, or your connection might be too slow. There are many other perfectly reasonable reasons for you not to put your camera on and you should not feel pressured to do this. If you simply say "Sorry, I can't turn my camera on today" then I won't ask any further and I will never explicitly ask you to turn it on.

18.2.3 Being appropriate

You should already be treating online meetings like physical ones e.g. turning up on time, being prepared, listening, engaging etc. Similarly, if people can see you then you should ensure you are wearing appropriate clothes (wearing clothes is the absolute minimum here!) and in an appropriate place (the bathroom is probably not appropriate) as you would for a physical meeting.

18.2.4 Respecting others

If other people have decided to turn their cameras on then I ask that you show them respect by not recording anything without explicit permission. I won't touch on the legality of this as I believe that basic respect for each other should be enough to prevent any issues. You will take part in larger meetings where recording may be standard and in such cases this should be made explicit.

Thanks to Giles Reger and Sarah Clinch for the text above

Bibliography

Agnew, H. (2016). Big four look beyond academics: Firms want to make offers based on potential, rather than personal circumstance. *Financial Times*.

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.

Ashcroft, R. (1997). The drugs don't work. In Potter, C., editor, *Urban Hymns*. The Verve, Hut Records. https://www.youtube.com/watch?v=ToQ0n3itoII.

Assay, M. (2020). Want an IT job? look outside the tech industry. *TechRepublic*.

Babineaux, R. (2013). *Fail Fast, Fail Often: How Losing Can Help You Win*. Tarcher.

Ball, C. (2020). What might the graduate labour market look like in 2021? *WONKHE*.

Barrass, R. (2002). *Scientists Must Write: A Guide to Better Writing for Scientists, Engineers and Students*. Routledge.

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.

Black, J. (2017). *Where am I Going and Can I Have a Map? How to take control of your career plan - and make it happen*. Robinson.

Black, J. (2019a). Dear jonathan: Careers column. *The Financial Times*.

Black, J. (2019b). How to write a top-notch CV. *Financial Times*.

Bolles, R. N. (2019). *What Color Is Your Parachute? A Practical Manual for job-hunters and career-changingers*. 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. *Financial Times*.

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 Mocine-McQueen, J. (2019). *How Your Story Sets You Free*. Chronicle Books.

Brewton, S. (2014). Squire bill widener vs. theodore roosevelt.

Britton, B. (2019). No sexuality please, we're scientists.

Brown, G. (2021). *How to get your PhD: A Handbook for the Journey*. Oxford University Press.

Brown, N. and Wilson, G. (2018). Ten quick tips for teaching programming. *PLOS Computational Biology*, 14(4):e1006023.

Caroll, L. (1865). *Alice's Adventures in Wonderland*. Macmillan.

Carr, J. (2018). 'it's not a bug, it's a feature.' trite—or just right?

Charette, R. N. (2005). Why software fails. *IEEE Spectrum*, 42(9):42–49.

Coughlan, S. (2019a). 'grade inflation' means 80% more top degree grades. *BBC News*.

Coughlan, S. (2019b). The symbolic target of 50% at university reached. *BBC News*.

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.

Dale, S. (2018). #NotJustLondon — what's that then? *21st Century Mindset*.

Damore, J. (2017). Google's ideological echo chamber.

David, J. (2017). How long should your CV be? *cv-library.co.uk*.

Davidova, P. (2021). My worst software engineering interview fails: Failing my facebook and google interviews.

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. *osmosis.org*.

Dweck, C. (2014). The power of not yet. *TEDxNorrköping*.

Dweck, C. (2017). *Mindset: Changing The Way You think To Fulfil Your Potential*. Robinson.

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.

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.

Fincher, S. and Finlay, J. (2016). *Computing Graduate Employability: Sharing Practice*. Council of Professors and Heads of Computing CPHC.

Fogarty, T. (2015). Hackathons are for beginners. *medium.com*.

Forever, B. (2019). Ten reasons why books are our best friends. *The Times of India*.

Friedman, S. and Laurison, D. (2020). *The Class Ceiling: Why it Pays to be Privileged*. Policy Press.

Fry, S. (2018). *Heroes: The myths of the Ancient Greek heroes retold*. Penguin Random House UK.

Garone, E. (2014). To print or not to print — a cv, that is. *bbc.com worklife*.

Gill, E. and Statham, N. (2021). Head who promised stability for ’forgotten school’ with nine leaders in 10 years is leaving. *Manchester Evening News*.

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. 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. *medium.com*.

Googler, A. (2019). Google technical writing courses: Every engineer is also a writer.

Graham, P. (2005). Good and bad procrastination. *paulgraham.com*.

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.

Hacker, A. (2021). International Collegiate Programming Contest (icpc). *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. *Waterstones*.

Haig, M. (2019). *Notes On A Nervous Planet*. Canongate Books Ltd.

Hamedy, R. (2019). How to build a reputable StackOverflow profile. *medium.com*.

Hendrix, J. (1967). Are you experienced? In Chandler, C., editor, *Are You Experienced?* The Jimi Hendrix Experience, Track Records. https://www.youtube.com/watch?v=s4oZdUV-G-Y.

Hitchens, C. (1997). Everyone has a book inside them. *Cable-Satellite Public Affairs Network (C-SPAN)*.

Hoffman, R. (2020). Customize your public profile URL. *linkedin.com*.

Hull, D. (2020). Getting started with latex: A laboratory manual. *latex4year1.netlify.app*.

Investor, A. (2019). Let's make work better. *investorsinpeople.com*.

Investor, A. (2020). How to bring yourself to work and allow others to do the same. *investorsinpeople.com*.

Isley, O., Isley, R., and Isley, R. (1988). Shake your thang. In *A Salt with a Deadly Pepa*. Salt-N-Pepa, London Records.

Johnson, H. (2020). The 'forgotten school' with nine head teachers in 10 years - and the man with a plan to turn it around. *Manchester Evening News*.

Johnstone, M. (2012). I had a black dog, his name was depression. *World Health Organization*.

Kellett, T., Taylor-Firth, R., and Boyle, R.-A. (1997). You're not alone. In Kellett, T. and Taylor-Firth, R., editors, *Extra virgin*. Olive, RCA Records.

Knuth, D. E. (1984). Literate programming. *The Computer Journal*, 27(2):97–111.

Kottke, J. (2018). Alan turing was an excellent runner. *kottke.org*.

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. (1979). 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.

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. https://www.youtube.com/watch?v=usNsCeOV4GM.

Leslie, I. (2020). Why your 'weak-tie' friendships may mean more than you think. *bbc.com worklife.*

Lewis, P. (2017). 'i see things differently': James damore on his autism and the google memo. *The Guardian.*

Malan, D. J. (2010). Reinventing CS50. In *Proceedings of the 41st ACM technical symposium on Computer science education - SIGCSE '10.* Assoication for Computing Machinery.

Malan, D. J. (2017). CS50 at scale. *YouTube.com.*

Malan, D. J. (2020). Teaching from home via zoom. *medium.com.*

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.

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.*

Markow, W., Coutinho, J., and Bundy, A. (2019). Beyond tech: The rising demand for IT skills in non-tech industries. *Burning Glass.*

Martin, R. C. (2011). *The Clean Coder: A Code of Conduct for Professional Programmers.* Prentice Hall.

McDowell, G. L. (2011). Ex-googler reveals strategies to land interview with google. *cnbc.com.*

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.

Meisler, B. (2012). The real reason silicon valley coders write bad software. *The Atlantic.*

Milliken, S. (2019). *From Learner to Earner: A recruitment insider's guide for students wanting to achieve graduate job success.* Rethink Press.

Newton, I. (1675). Letter to robert hooke. *Historical Society of Pennsylvania.*

Ong, J. and Lopez, L. (2019). Create your resume for google: Tips and advice. *Google.*

Parker, P. (1962). With great power comes great responsibility. *wikipedia.org.*

Petroski, H. (1992). *To Engineer Is Human: The Role of Failure in Successful Design.* Vintage.

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.* BBC.

Raymond, E. S. (1999). *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* O'Reilly Media.

Richards, S. (2019a). *Content Design London: Readability Guidelines*, chapter Why people do not read online. readabilityguidelines.co.uk.

Richards, S. (2019b). *Content Design London: Readability Guidelines*, chapter Treat links with respect. readabilityguidelines.co.uk.

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.

Roberts, R. J. (2015). Ten simple rules to win a nobel prize. *PLOS Computational Biology*, 11(4):e1004084.

Ryder, S. (1988). Wrote for luck. In Hannett, M., editor, *W.F.L. Think About the Future Mix.* Happy Mondays, Factory Records.

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.

Santos, L. (2020). Five favorite coping tips for world mental health day. *The Happiness Lab.*

Santos, L. (2021). The science of well-being at yale university.

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. *bfi.org.uk*.

Sedgwick, R. (2019). Should computer science be required? *Inside Higher Ed*.

Shaha, A. (2014). Wave machine demonstration.

Shakespeare, W. (1597). *Romeo and Juliet*.

Shimer, D. (2018). Yale's most popular class ever: Happiness. *The New York Times Magazine*.

Smith, M. E., Smith, B., and Rogers, S. (1987). Hit the north. In *The Frenz Experiment*. The Fall, Beggars Banquet.

Stanford-Clark, A. (2019). Innovation begins at home. *Alliance Manchester Business School*.

Swinton, J. (2019). *Alan Turing's Manchester*. Infang Publishing.

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.

Tyldum, M. (2014). The imitation game.

UK, G. (2020a). Discrimination: your rights and protected characteristics. *www.gov.uk*.

UK, G. (2020b). Handing in your notice. *www.gov.uk*.

UK, G. (2021a). Disciplinary procedures and action against you at work. *www.gov.uk*.

UK, G. (2021b). Dismissal: your rights. *www.gov.uk*.

UK, G. (2021c). Raise a grievance at work. *www.gov.uk*.

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.

Wax, R. (2014). *Sane New World*. Hodder.

Way, M. (2017). What I cannot create, I do not understand. *Journal of Cell Science*, 130(18):2941–2942.

Wickham, H. and Grolemund, G. (2017). *R for Data Science*. O'Reilly UK Ltd.

Woodruff, W. (2003). *Beyond Nab End.* Abacus.

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.