Dr. Duncan Hull

Yours truly

Contents

Hello

Figure 1: Hello my name is Duncan, this is what I look like now (left) and when I was a longer-haired undergraduate (right). I'm a lecturer here in the Department of Computer Science at the University of Manchester.

see https://gist.github.com/dullhunk/45de6737e7cff0997c131acf8a1152ed

Hello and welcome to the Department of Computer Science at the University of Manchester. My name is Duncan and I'm a lecturer (Assistant Professor) here with responsibility for managing our Industrial Experience (IE) program. I teach undergraduate & postgraduate students, supervise tutorials & projects while serving as industrial experience & employability tutor and departmental forum chair. I'm interested improving teaching, learning and the student experience by:

- Developing and delivering *Coding Your Future*: a course & guidebook for students at cdyf.me
- Teaching engineers to make better software in collaboration with our industrial mentors

- Updating students weekly on opportunities with the Wednesday waggle newsletter at waggle.cs.manchester.ac.uk
- Improving students written communication skills by enabling them to start writing their future
- Engaging with a wide range of employers through our industry club
- Growing and building the teaching community by journal clubbing at sigcse.cs.manchester.ac.uk
- Supporting local schools teaching computing by coding their future

Our elective industrial experience program has up to 100 students every year working in paid employment for the penultimate ("sandwich") year-in-industry of their undergraduate degrees. If you are an employer who would like to recruit a summer intern, placement student or graduate please get in touch.

Full stack teaching

Regardless of the age or the stage, I enjoy the challenges of teaching and have taught english, maths, science and engineering to primary & secondary school children, undergraduates & postgraduates. In 2011, I completed a Postgraduate Certificate in Education (PGCE) at the University of Bath and worked as a high school science teacher in co-educational comprehensive (non-selective) state schools in Swindon, Shaftesbury and Stockport before returning to higher education. As well as teaching in the UK I have been fortunate enough to teach in India, Japan and America too.

What's the story?

What's the story, Jackanory? Born in Bath, Somerset and raised using a secret West Country recipe, my story is a mixture of Natural Science (Plant Sciences, BSc), Computer Science (MSc & PhD) and software engineering. Outside of academia, I've worked as a consultant and software developer for BBC Monitoring, the Ford Motor Company and the National Health Service (NHS). As an academic, I have been part of the e-Science lab, Apache Taverna and myGrid projects and completed a postdoc at the Manchester Institute of Biotechnology (MIB) on the Pathtext project. This was followed by a short stint as a software engineer of Chemical Entities of Biological Interest

(ChEBI) at the European Bioinformatics Institute (EBI) in Cambridge, UK.

Toolbox

Last updated on 24 September, 2022, this website was built using bookdown with R markdown, the R language, JavaScript, knitr, LaTeX, Pandoc, RStudio and TLC. The source is available on github and the documentation in *Authoring Books and Technical Documents with R Markdown*. These pages are also available in one single pdf file and and an ebook. Thanks to Yihui Xie and his collaborators for the handy tools and excellent documentation.

Get a life!

When I get the time I like to sing bass-baritone, look after bees and butcher the greek language, preferably on a beach in Greece. Α !

Chapter 1

Students

I teach, mentor, tutor, lecture on and supervise a variety of undergraduate and postgraduate courses. You can find me online during office hours, in the labs, my office, on YouTube and in the lecture theatre.

Figure 1.1: Posing on the BBC Breakfast red sofa with the winning team of the BBC / Barclays University Technology Challenge (UTC) in MediaCityUK, Salford

1.1 All years: Debug your future

If you'd like to debug your CV, application form, covering letter and job search etc, read debugging your future (Hull, 2022b) and hacking your future (Hull, 2022c), especially if you haven't written a CV, Résumé or LinkedIn

profile before. Once you've done this you can send me any relevant documentation and then:

- Drop-in to my weekly one-to-one CV clinics for Computer Science students online during term-time during my open office hours
- Get feedback on your CV from as many other people as possible, because "given enough eyeballs, all bugs are shallow" (Raymond, 1999)

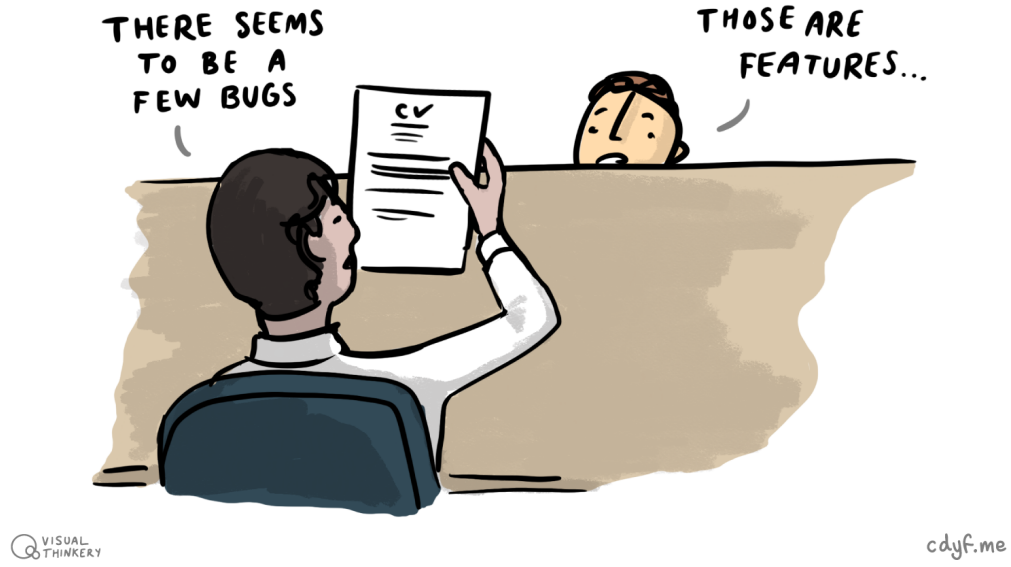

Figure 1.2: Is that a bug or a feature in your CV? It's important to debug your CV before an employer sees it, see the Wednesday Waggle for details of my weekly live debugging sessions. Features not bugs picture by Visual Thinkery is licensed under CC-BY-ND

Outside of term time, it's best to book a debugging appointment.

1.2 First year students

If you're in your first year of study, I serve as:

- Academic staff member for First year team projects: COMP101 led by Ulrike Sattler (Sattler, 2022), see the getting started with LaTeX lab

manual
- Mentor and tutor to one group of six first year students
- Organiser of first year guest lectures, which mostly run in the second semester, February to May

1.3 Second year students

If you're in your second year of study, I serve as:

- Course leader for Coding your Future: COMP2CARS a course designed by me for (primarily) second year students to design, build, test and debug their futures in computing.
- Organiser of the labs for the software engineering mentoring program
- Teaching team member for second year software engineering: COMP23311, a course designed by Suzanne Embury (Embury, 2022; Eraslan et al., 2020)

Previously I served as second year tutor from 2016 through to 2020, and course leader for second year software engineering from 2020 to 2022.

1.4 Penultimate year students

If you're on placement during your penultimate year (your last-but-one), I serve as:

- Course leader for "with industrial experience" (IE), an elective and intercalated year in industry.
- Year tutor for students on placement, either face to face or via telecon

1.5 Final year students

If you're in your final year of study, you'll find me:

- Supervising final year educational projects based in secondary schools in Greater Manchester, see coding their future. (Hull, 2020a)

1.6 Masters students

If you're doing a Masters degree, you'll find me:

- Leading the course on Principles of Digital Biology which covers Bioinformatics, Computational Biolgy and a bit of medical informatics
- Supervising Master of Science projects in Computer Science and Data Science. (Wickham and Grolemund, 2017) This typically involves various combinations of Wikipedia, Wikdata, SPARQL (DuCharme, 2013) and chatbots. (Sharwood, 2019)

1.7 PhD students

If you've got any interesting PhD research ideas you like to pitch to me, particularly if they involve computer science education or professional issues, I'm all ears!

1.8 Extracurricular

Outside all of the above, I also get involved in:

- Organising, facilitating and promoting extra-curricular activities such as hackathons (Briscoe and Mulligan, 2014; Warner and Guo, 2017) and edit-a-thons. (Hull, 2017, 2015; Mohammad-Qureshi and Hull, 2019) These usually occur off-timetable, for example Wednesday afternoons, evenings and weekends.
- Judging competitions such as studenthack.com and greatunihack.com since 2014. These two hackathons are organised by UniCS, a student-led tech society formed by the merger of HackSoc (computer geekery) and CSSoc (socialising). Many other hackathons exist, they are usually aimed at beginners looking to learn new skills, rather than experts looking to compete. (Briscoe and Mulligan, 2014; Fogarty, 2015)

Chapter 2

Employers

We work with a wide range of employers from the smallest bedroom startup to the worlds largest multi-national corporations, and are always looking for more organisations that can offer our students a stimulating working environment. According to highfliers.co.uk, the University of Manchester is the most targeted University in the UK by the Times Top 100 Graduate Employers. (Birchall, 2020a, 2021, 2020b, 2019) We can still do better, for example by engaging with a more diverse group of employers, especially those in Manchester and the Northern Powerhouse, see git.io/manc. (Hull, 2020b; Davis, 2014b,a; Ovenden, 2019; Wainwright, 2019)

2.1 Recruiting students

If you are recruiting computer scientists and software engineers as a summer interns, placement students or as graduates please get in touch with me or Mabel Yau (careers and placements officer). We typically have around 250 undergraduate students graduating annually, alongside a smaller number of Masters and PhD students. The entry tariff of our students (A* A* A* including mathematics) is comparable to other leading Computer Science (CS) departments in Russell Group universities as shown in the table below.

Institute	UCAS entry tariff
University of Manchester	A* A* A*

Institute	UCAS entry tariff
University of Cambridge	A* A* A
University College London	A* A* A
Imperial College London	A* A* A
University of Oxford	A* A A

If you are looking to recruit science and engineering students from other disciplines like Physics, Maths, Chemistry, Mechnical, Aerospace & Civil Engineering (MACE), Materials Science and Electrical & Electronic Engineering (EEE) you should talk to:

- academic staff in those departments and/or
- the central careers service of the University via employers.manchester.ac.uk

2.2 Careers fairs

Our annual Computer Science careers fair is held in the Kilburn building in autumn, we typically have around 30 employers exhibiting over two days. As space is limited, we are always over-subscribed and are not able to accommodate every employer that our students will be interested in. We give priority to employers that offer internships, placements and graduate roles and have contributed to our community through the activities described on this page. The central careers service also organises:

- the big careers fair every autumn, see the events
- a smaller careers fair in May
- hundreds of other employer events on campus during term time (Birchall, 2020b)

2.3 Drop-in sessions

If you aren't willing or able to exhibit at careers fairs, we also run ad-hoc drop-in sessions where employers can come in and set up a stand in the foyer to talk to computer science students informally on their way to and

Figure 2.1: Any employer recruiting our students is welcome to join our industry club, see details below

from lectures. These usually happen during lunch in term time. If you're interested in exhibiting at either of these events, please contact the careers and placements officer Mabel Yau.

2.4 Industry Club

All employers are welcome to join our industry club mailing list by sending an email to listserv@listserv.manchester.ac.uk with the the text **subscribe cs-**

industryclub yourfirstname yoursecondname in the body of the email message. The industry club is part of our wider business engagement activities.

The mailing list is low-traffic, typically two to three updates per year and an invitation to our annual industry club meeting. We promise not to spam you or sell your email details on to third parties.

2.5 Industrial mentoring

The Industrial mentoring scheme for software engineers allows employers meet students during code review sessions.

2.6 Co-supervised projects

If you would like to co-supervise a project student in collaboration with an academic member of staff, there are several options. The best option depends on the domain, level and duration of the project:

- **Bachelors projects**: these are completed in the final year of a Bachelors degree and last for six months, starting in September and finishing in March. Projects are proposed (and offered to students) in March and start in September of the same year.
- **Masters projects**: again these are six months in duration but start in March and finish in September. Projects are proposed (and offered to students) in the preceding November.
- **PhD projects**: For industrially sponsored or co-supervised projects, speak to the research office at cs.manchester.ac.uk/research.
- **Knowledge Transfer Partnerships**: We have a range of KTPs, speak to the research office for details
- **Impact Acceleration Accounts**: We have a range of IAAs, speak to the research office for details

For Bachelors and Masters projects, you can contact academic members of staff directly, or speak to Tim Morris (final year project lead) or Caroline Jay, who leads our postgraduate taught (Masters) courses.

2.7 The Wednesday Waggle

During term time, we highlight events and vacancies for Computer Science students from a wide range of sources in a weekly newsletter called the *Wednesday Waggle* (waggle.cs.manchester.ac.uk) . This goes out to around ~1500 Bachelors and Masters Computer Science students in Manchester each week. If you have vacancies or events you would like our students to know about, you can advertise them at careerconnect.manchester.ac.uk, and let me know when they are live in the system, then I can highlight them to the weekly newsletter. Alternatively, point us to where they are advertised online, see advertising vacancies to students. It can help students find your needle in the jobs haystack.

2.8 Join the community

There is a thriving community of engineers and entrepreneurs in Manchester and across the North of England. One of the best ways to recruit engineers and scientists is to join and *contribute* to the community. Get involved in events, sponsor a hackathon, deliver a guest lecture, host your own event or become a software engineering mentor. Employers who engage **early and often** are much more likely to get something back. As an employer, you may also be interested in events run by:

- The Institute of Student Employers (ISE)
- The Association of Graduate Careers Advisory Services (AGCAS)
- The Work Based and Placement Learning Association (ASET)

If you're a startup new to employment, you may find the guide at gov.uk/employ-someone useful.

2.9 Buzzing!

At peak times, we can get **very busy** with many concurrent employer events on campus, see figure 2.2. Please be patient and persistent if we do not reply immediately. Unfortunately, we are not always able to respond to everyone because our students, staff and space are all finite resources. We give priority to employers that have already given their time and expertise to our community.

Table 4.6 Universities Targeted by the Largest Number of Top Employers in 2021-2022

	Ranking in 'Good University Guide' *		Ranking in 'Good University Guide' *
1. Manchester	23	11. Bath	9
2. Nottingham	28	12. London University College	7
3. Bristol	14	13. Durham	6
4. Birmingham	25	14. Exeter	21
5. Leeds	15	15. London Imperial College	4
6. Warwick	8	16. London King's College	18
7. Cambridge	3	17. Southampton	16
8. Sheffield	22	18. Glasgow	12
9. Edinburgh	13	19. London Queen Mary	40
10. Oxford	2	20. London School of Economics	5

Source - The Graduate Market in 2022

* *The Times & Sunday Times Good University Guide 2022*

Figure 2.2: According to highfliers.co.uk, the University of Manchester is the most targeted University in the UK by the Times Top 100 Graduate Employers (Birchall, 2022)

2.10 Employability

We are working hard to improve the employability of students because while having a Computer Science is necessary for some jobs, it is not sufficient. (Anonymous, 2013; Shadbolt, 2016; Fincher and Finlay, 2016; Fincher et al., 2017) Over the last decade we have been successful in *more than doubling* the number of our students going on year long placements in industry to around 100 per year, see figure 2.3. This is a win-win-win situation for:

1. **Students**: benefit from a broader education, and develop social and non-cognitive skills that can be challenging to teach and learn in a purely academic environment. This is known as the winning personality (de Vries and Rentfrow, 2016).
2. **Employers**: placements are a cost-effective way for employers to recruit (and retain) graduate talent
3. **Universities**: produce better graduates (Mandilaras, 2004) with broader and deeper skills, who earn more and get better jobs (de Vries and Rentfrow, 2016). Well paid placements can also facilitate social

mobility. (Wang and Crawford, 2018)

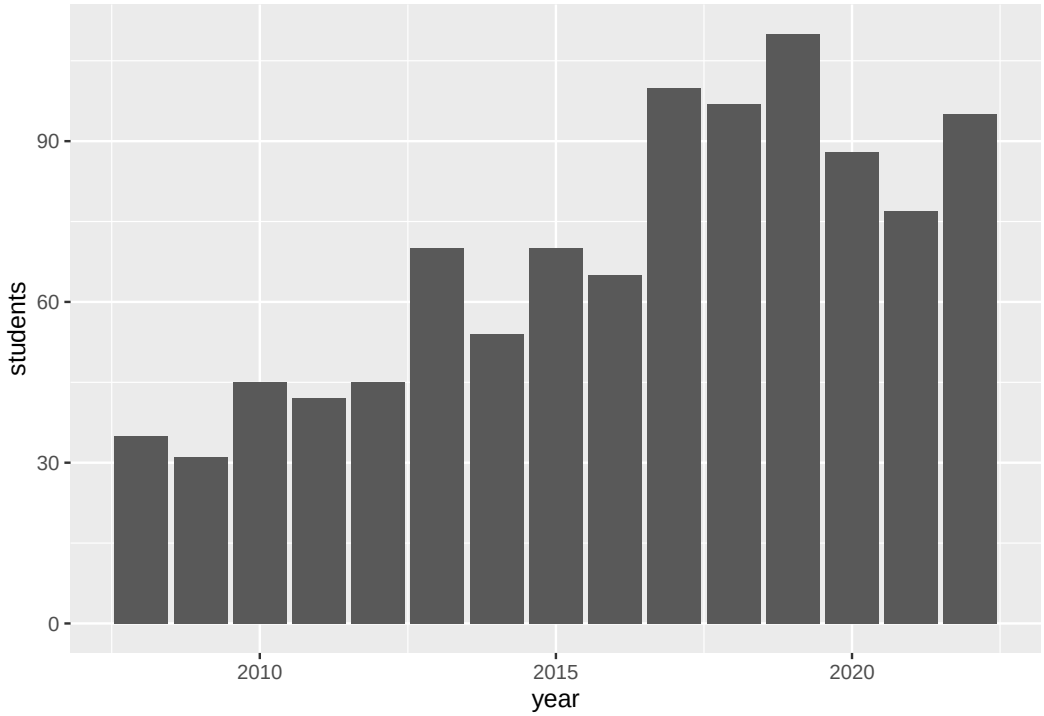

Figure 2.3: Number of undergraduate Computer Science students completing a year in industry as part of their degree at the University of Manchester. Since 2008 over 900 students have completed the program. I have been leading the program since 2012 and managed to ~double the number of students doing placements per year. As you can see in the histogram, the COVID-19 pandemic started having an effect on students starting placements in 2020.

Previously our students have secured year long placements at a wide range of employers including Accenture, Agilent Technologies, Amazon, AND Digital, Apadmi, Arggo, ARM, Autodesk, AVL Powertrain, BAML, the BBC, Biorelate, BJSS, Bloomberg, BMW Mini, Bsquare Controls, BT, Cantarus, Celtra, CERN, Codethink, d3t, Elysian Systems, Feral Interactive, Fidelity, FiveAI, HMRC, IBM, Imagination Technologies, Intel, ISA Software, JP Morgan, Keysight Technologies, KPMG, Matillion, McAfee, Mentor Graphics, Monoprix, Morgan Stanley, NCC Group, Nokia, Nomura, Novacoast, Ocado, PA Consulting, PwC, Schlumberger, ServiceNow, Siemens, Soda Software, SteamaCo, The Hut Group, The Start Up Factory, Uber, Visa and Vodafone.

There's still more we can do to improve the employability of our graduates. If you'd like to help our graduates become more employable, get in touch.

Chapter 3

Research

My research interests are in Computer Science Education and pedagogy. (Fincher and Robins, 2019; Biggs and Tang, 2011; Fry et al., 2014) I'm interested in methods that can improve learning and student experience using techniques like coding your future, journal clubbing, industrial mentoring, live music, working with schools, editing Wikipedia and more.

Figure 3.1: Too many educational practices are not backed up by good evidence that they actually work. More evidence is needed to support many of the claims made about effective pedagogy. *Wikipedian Protester* cartoon by Randall Munroe at xkcd.com/285 published under a Creative Commons Attribution-NonCommercial 2.5 License

3.1 SIGCSE

Computer Science has only been taught to undergraduates in the UK for 50 short years (Brackenbury, 2005; Hawthorne et al., 2019), so there's lots of open questions about how to teach the practical, theoretical and professional aspects of the subject. To that end:

- I'm an active member of the Association for Computing Machinery (ACM) and it's Special Interest Group (SIG) in Computer Science Ed-

ucation (sigcse.org). In 2020 I founded the ACM SIGCSE journal club and chair the monthly Manchester meetup. Anyone is welcome to join, see sigcse.cs.manchester.ac.uk/join-us

- I serve on the program committee of the United Kingdom and Ireland Computing Education Research (UIKICER) conference, on the board of UK ACM SIGCSE and have served on the program committee for Computing Education & Practice (CEP) conference at Durham University

3.2 Industrial mentoring

Since we started the industrial mentoring scheme for software engineers in 2015, more than 1000 students have been through the mentoring scheme with 250 students taking the course every year. We are very grateful for continued support from our industrial partners in making this happen.

Mentors meet with a group of six second year students for two one hour meetings and do some gentle code review of their gitlab repository, as they start to fix bugs and add features to a large open source software project. You don't *need* to be an expert in the tools students are using (Java, Eclipse, Jenkins, Git, JUnit and Ant) it is more about the general process (and politics) of building and testing high quality software in large and distributed teams, than the specifics of the stendhalgame.org codebase we happen to be using. Mentors are typically software engineers, both junior and senior.

3.3 Coderdojo & Code Club

I'm a volunteer at coderdojo.com (Ward, 2016). Coder dojos are local community engineering clubs for young people; with several other volunteers I help out at CoderDojo North West. We meet once a month to help young people broaden their digital and computational horizons.

Previously I lead an after school CodeClub as part of a global network of free coding clubs for 9–13 year olds. (Smith et al., 2014) As with coderdojo, the aim is to have fun using Scratch, (Resnick et al., 2009) python and other interesting technology we can get our hands on including Raspberry Pi, (Halfacree, 2020) Micro:bits, (Sentance et al., 2017) LEGO® MINDSTORMS®,

(Papert, 1980; Klassner and Anderson, 2003) Oculus Rift, Sonic Pi (Aaron et al., 2016) and CodeBug etc.

3.4 Wikipedia

Wikipedia and wikidata.org (Vrandečić and Krötzsch, 2014; Turki et al., 2019) are powerful tools for improving both digital skills and communication skills, regardless of your age or level of computer literacy, (Proffitt, 2018; Reagle Jr., 2010; Littlejohn et al., 2019) particularly in the following areas:

- Literacy generally, the ability to read and write in any natural language. The literacy skills of some engineers and scientists leaves plenty of room for improvement, but literacy has many overlapping dimensions including:
  - Data literacy the ability to read and write (data)
  - Digital literacy the ability to read and write (digitally)
  - Computer literacy the ability to read and write (using a computer)
  - Information literacy the ability to read and write (information)
  - Scientific literacy the ability to read and write (science). How many people do you know who *unashamedly* proclaim their scientific or mathematical illiteracy? (Stacey, 2009; Gowers, 2016; Garner, 2012)

As an experienced and long serving editor of Wikipedia since 2004, I organise and participate in Wikipedia training events which recruit new Wikipedia editors. Some recent examples include:

1. 2020-06-24 Wikipedia: Women, War and Peace run in collaboration with the Imperial War Museums' War and Conflict Subject Specialist Network, with support from the Arts Council England and Art Fund.
2. 2020-02-26 Wikimedia in Education UK Summit, Coventry University #wikiedu20
3. 2019-11-22 Training of Trainers (ToT) workshop, University of Glasgow
4. 2019-10-19 Learn to edit Wikipedia with Ada Lovelace, Sackville Street Building, University of Manchester (Mohammad-Qureshi and Hull, 2019)
5. 2019-10-12 Wikipedia Edit-a-Thon with Zebra Hub HQ, Pankhurst Centre, Manchester

6. 2017-10-13 Physiology Friday, Hodgkin Huxley House, Farringdon, London (Hull, 2017)
7. 2015-09-02 First Wikipedia Science Conference #wikisci, Wellcome Collection, London, NW1 (Hull, 2015; Hodson, 2015)

More information on past and future events like this can be found at:

- wiki-loves-scientists.org.uk
- en.wikipedia.org/wiki/User:Duncan.Hull

3.5 Tuning complete

Tuning complete are a musical collective from Manchester, Lancashire , named after the famous Computer Scientist Alan Tuning. We use his eponymous Tuning machine to make music which is quality assured using the Tuning test.

Our membership is fluid but our founding members, some of whom can be seen in figure 3.2 include:

- Jez Lloyd: Bachelor of Music, DJ and backing vocals
- Steve Furber: bass guitar.
  - "All we've got is Steve's guitar, three chords and the truth." (Howard, 1951; Dylan and Hewson, 1988)
- Justin Timberfake: lead vocals, lead dancer (Timberlake et al., 2016)
- Billie Fakish: guest vocalist (Eilish and O'Connell, 2019)
- Olivia Fakerigo: guest vocalist (Rodrigo et al., 2021)
- Duncan Hull: MC,
  - synth (MicroKORG)
  - drum machine (Sonic Pi)
  - embarrassing dad dancing (Facepalm, 2015)

Theoretically, we are a Turing Complete band. (Turing, 1937; Brailsford, 2005) Artistically, this means that what we lack in youth, good looks, fame, fortune, fashion sense, fanbase and back catalogue we compensate for with:

Musical geekery (Fauvel et al., 2006; Harkleroad, 2006)
Mathematical geekery (Rosenthal, 2005)
Computer geekery (Aaron et al., 2016)

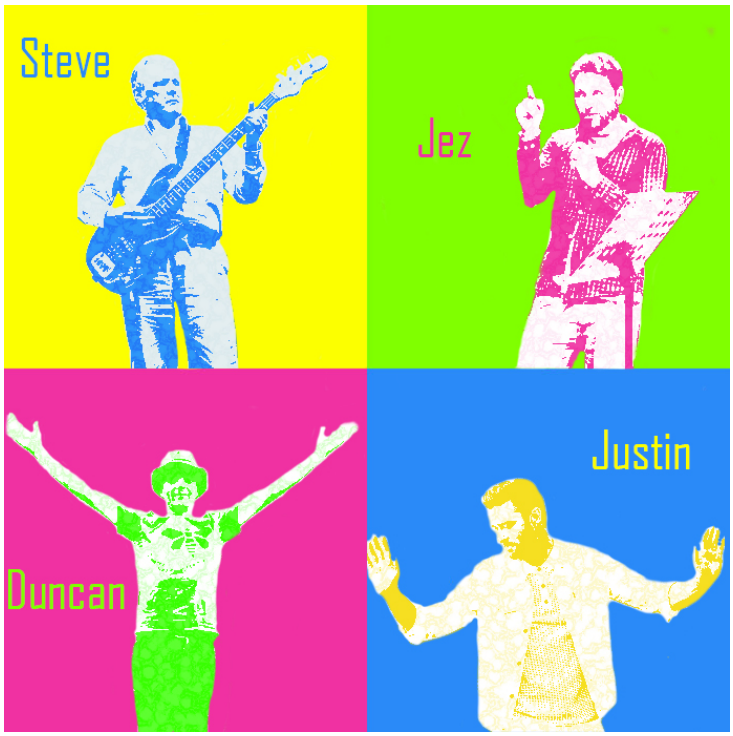

Figure 3.2: The founder members of Tuning Complete were Jez Lloyd, Steve Furber, Justin Timberfake and yours truly.

We played our debut gigs to packed theatres of over 200 second year & first year undergraduate computer science students in the autumn of 2019 and are currently planning future live events while writing a (hopefully) lucrative hit single, working title: #LivingTheDream. If you would like to book our services for your next event, hackathon, wedding, bar mitzvah etc, please contact our agent Mrs. Kilburn shown in Figure 3.3.

Figure 3.3: Mrs. Kilburn is our manager, booking agent and promoter. She is the power behind our boy band throne, so all bookings must be approved and scheduled by her office. Please do not approach band members directly with gig requests or offers of marriage, we are all answered for!

3.6 Publications

Informal publications can be my lab log:

- duncan.hull.name/lablog

Formal peer-reviewed publications can be found on DBLP, ORCID, Google Scholar, the ACM Digital Library, Wikidata etc:

- dblp.org/pid/h/DuncanHull
- wikidata.org/wiki/Q47012855
- dl.acm.org/profile/81350580198

- orcid.org/0000-0003-2387-503X
- europepmc.org/authors/0000-0003-2387-503X
- profiles.impactstory.org/u/0000-0003-2387-503X
- scholar.google.com/citations?user=iDJ-t7IAAAAJ

According to Google scholar, my most cited papers are on:

1. Apache Taverna, published in *Nucleic Acids Research* (Hull et al., 2006)

2. Another Taverna paper, published in *Concurrency and Computation* (Oinn et al., 2006)

3. A paper on modelling human metabolism, published in *Nature Biotechnology* (Thiele et al., 2013)
4. A review of tools for managing large bibliographies, published in *PLOS Computational Biology* (Hull et al., 2008)

The first paper for which I was formally acknowledged was on simulated environmental change in the subarctic published in *New Phytologist* (Potter et al., 1995). I was a humble field assistant, not a co-author, one of the *absolut* best summer jobs I've ever had!

Chapter 4

Coding your future

Coding your Future is a guidebook written to help students design, build, test and code their futures in computing, see figure 4.1. Written by yours truly, (Hull, 2022a) the full text of the book is freely available online at www.cdyf.me. If you'd prefer to read it in ebook format, you can also download it in a single file, see section 4.2.

4.1 Reading your future

Coding your future is a guidebook and course is aimed at ALL students in higher education. While the guide supports undergraduate teaching at the University of Manchester, it doesn't actually matter:

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

Coding your Future has something for any student of computing, both those inside and outside of Computer Science departments.

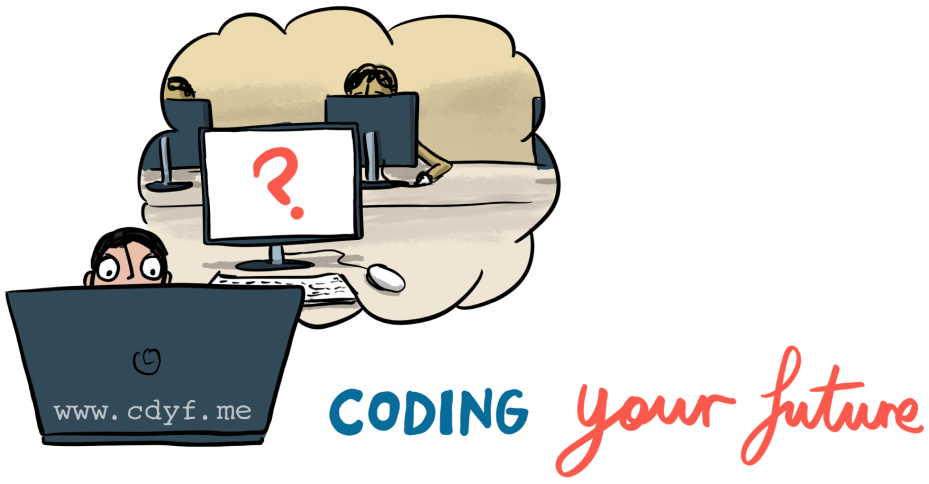

Figure 4.1: Coding your Future is a guidebook to help students design, build, test and code their futures in computing, see www.cdyf.me. *Coding your Future* illustration by Visual Thinkery is licensed under CC-BY-ND

4.2 Downloading your future

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

- cdyf.pdf
- cdyf.epub

Since the guidebook is published open access and open source, I welcome constructive feedback from students, employers, academics and any other contributors, see cdyf.me/#contributing

Chapter 5

Coding their future

Coding their future is a collaboration & partnership between secondary schools and the Department of Computer Science at the University of Manchester. Our aims are to:

- improve and support Computer Science education at key stages 3, 4 and 5. (Furber, 2012, 2017; Sentance et al., 2018; Swan et al., 2013)
- widen participation in higher education, especially in under-represented groups. (Agnew, 2020; Friedman and Laurison, 2020; Green and Kynaston, 2019; Rajan, 2019)
- enable our undergraduate students to develop their leadership and communication skills

Figure 5.1: Undergraduate students in computer science regularly work with schools as part of our wider social responsibility activities (Rothwell, 2011b,a) and schools, colleges and public activities

The University provides schools with a final year student who can teach Computer Science in your school or college as a teaching assistant (TA). In return, the school provides our undergraduate students with a safe and supportive environment in which to teach which extends and augments your current curriculum. This can either be an after school, extension / lunchtime club or during scheduled lesson time, typically between year 7 and 13. This is similar to the Undergraduate Ambassador Scheme (UAS), (Singh, 2005; Cooper and D'Inverno, 2005) and school placements (Moller and Powell, 2019) except students work is assessed using our final year project framework. (Morris, 2019a,b) Since these projects were started in 2012, our students have worked with a range of schools in the private and public sector, both selective and non-selective, co-educational and single-sex including:

- Fairfield High School for Girls, Droylsden
- Trinity CofE High School, Central Manchester
- University Technical College (UTC) @MediaCityUK, Salford
- Manchester Communication Academy, Harpurhey

- The Barlow RC High School, Didsbury
- William Hulme Grammar School, Whalley Range
- Cheadle Hulme High School, (CHHS) Stockport
- Laurus Cheadle Hulme, Stockport
- Knutsford Academy, Knutsford
- Altrincham Grammar School for Girls, Trafford
- Altrincham Grammar School for Boys, Trafford
- Manchester Grammar School (MGS), Fallowfield
- Stretford Grammar (SGS), Stretford

The projects were setup by Duncan Hull and David Rydeheard (who retired in 2019), and are now run and supervised by Duncan. We hope to transfer ideas between private and public sector, as there are lots of open questions about how Computer Science should be taught. (Sentance et al., 2018; Sentance, 2018; Stephenson, 2018; Fincher and Petre, 2004) To find out more, see the guidance for teachers and guidance for students below.

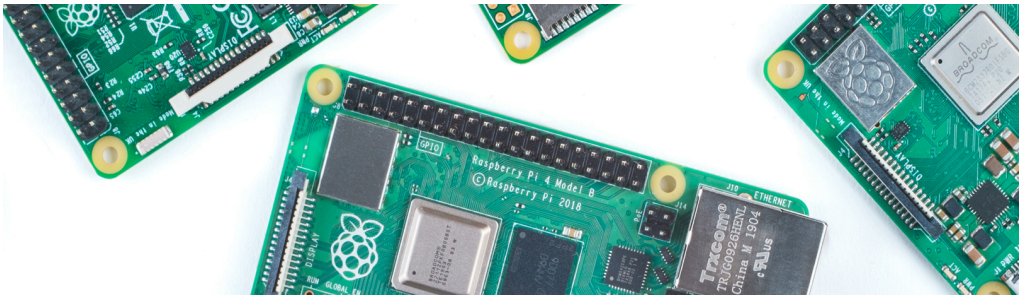

Figure 5.2: An abundance of free software and relatively cheap new hardware like the Raspberry Pi (Halfacree, 2020), Microbit, (Sentance et al., 2017) Makey Makey [Rogers et al. (2014); Shaw (2012);] Crumble Controller and Arduino (Banzi and Shiloh, 2015) has opened up lots of new possibilities for teaching Computer Science. Picture via Alex Bate. (Bate, 2019))

5.1 Guidance for teachers

Our aim is to support the teaching and learning of Computer Science in your school and to help engage schoolchildren in the subject. This page describes what we can provide you with and what we expect to get in return.

5.1.1 What the University is offering your school

The University of Manchester will provide your school or college with at least one student ambassador with some relevant training who has completed two years of study in Computer Science and has:

- A good knowledge of, and enthusiasm for Computer Science
- Completed Disclosure and Barring Service (DBS) clearance
- An interest in teaching and working with young people
- Achieved a minimum of a 2:1 or 1st class degree in their second year

5.1.2 What the University expects from your school

In return, we expect that the school provides the undergraduate student with:

- Opportunities to engage with a classroom or after school club of children as a Teaching Assistant (TA). This is typically for around one or

two hours during term time. Initially, this could be through classroom observation and teacher assistance, culminating in the student delivering at least one lesson (and potentially a series of lessons) with your support and guidance
- Advice, suggestions, feedback, assessment and encouragement from you to suggest the kinds of resources that would be useful, appropriate or engaging for the Computer Science curriculum you are teaching
- Classroom and behaviour management: the students are not trained teachers and will be relying on your expertise in classroom and behaviour management.

5.1.3 Resources developed by students

Undergraduates typically develop a range of resources. The project will involve development of a computer-based system together with supporting activities, lessons and resources. The resource could be a variety of things including, a game, robotics, animations, hardware (Raspberry Pi, Arduino etc) or software, intended to enthuse school students at one of the Key stages 3 or 4 about fundamental concepts in computing preferably linked to one of the new Computer Science curricula.

5.1.4 Project timing

The projects run for 6 months from September to March, divided into three phases.

1. **September to October** Observation in the classroom teaching by the student around once per week. Development of ideas for an educational tool that the student will make, with the advice of the classroom teacher
2. **November to January** From November to January, our students develop and tests prototype tool (or tools) with the supporting material, this can happen sooner for students who make a quick start to the project.
3. **February to April** From February to April, our students are expected to liaise closely with teachers to develop an educational tool that will be of use in the classroom using teachers' suggestions as to what is appropriate to build. Students will spend some time in a classroom

working closely with teachers and students developing and delivering a new resource for teaching. More details on final year projects can be found in COMP300, the undergraduates already know what is required from their project

5.1.5 Assessment and monitoring

Formal supervision and mentoring is undertaken by the university (Duncan and David), but we will ask you to fill in a one page form on your assessment of their progress during their time at your school, we very much value your input and hope that these projects can beneficial for both your school and the University. We don't want to burden you with unnecessary bureaucracy that all teachers battle with!

5.2 Guidance for students

Figure 5.3: Lecture theatre 1.1 (LT 1.1) in Kilburn full of first year students

So why would you, an undergraduate student, want to work on an education project in secondary school? The UK government would like Computer Science should be taught in all secondary schools in the UK. (Furber, 2017) However, in many UK schools there is a shortage of teachers who are trained in Computer Science, consequently, many teachers find themselves being asked to teach a subject they may know little about. (Furber, 2012)

Undergraduate students can make a significant difference here, by supporting teachers in the classroom to create and deliver new classroom resources in

Computer Science. (Hull, 2020a) In addition, undergraduate students will have the chance to:

- develop leadership skills in the classroom
- gain valuable experience of working on "real world" problems in a stimulating environment
- improve your communication skills, especially spoken communication work as part of a team (in the school) and join a small group of like-minded undergraduate students (in the University) working on related projects
- test your knowledge & technical ability in a challenging and dynamic environment working with young people
- last, but not least, there is a good chance you will have lots of fun and have a rewarding experience of teaching make yourself more employable by doing all of the above

5.2.1 Who is involved?

Initially, the number of undergraduate students involved in these projects will be less than ten. We also require that you will have a minimum of a 2:1 or 1st in your second year exams. Projects are co-supervised by Duncan with additional supervision from an experienced member of teaching staff at a participating school.

We have carefully selected schools in Manchester that are relatively easy for you to get to, are already teaching Computer Science and have supportive staff and teachers in place to help you. You will be expected to work directly with school children with the support of the teaching staff in your school. Schools we have worked with are all the Manchester area.

5.2.2 What will the educational projects be expected to deliver?

You will be expected to work closely with the teacher to develop resources that

- engage students with one or more aspects of the new Computer Science curriculum at an appropriate key stage. This is usually key stage 3, key stage 4 or key stage 5 ages 11-18.

- complement **and extend** the schools current provision for computer science in the school

During the project you will be spending a significant amount of time in the classroom, visiting your school every week during school term time throughout the duration of your project to develop resources. These must include a computer-based teaching tool which may use, for example, Raspberry Pi's, visual aids, demonstrations, videos, online questionnaires, formative feedback, games, drones, robotics, music, (Aaron et al., 2016) algorithms (Kubica, 2012) or even just the command line (Smedley, 2019) etc.[1] In addition, guidance on classroom use, such as a lesson or series of lessons to support the tool. Remember that you don't actually need a computer, see Computer Science Unplugged: Computer Science without a Computer. (Bell and Vahrenhold, 2018)

All deliverables for standard final year projects will be expected of these projects including:

- first semester presentation
- demonstration of the resource being used in the classroom
- final written report

Assessments for these projects will be as for standard projects, (Morris, 2019a,b) but part of the evaluation of the project will be a classroom demonstration, a description and evaluation of which should be included in your final report.

5.2.3 Blended learning

COVID 19 has changed teaching, from primary and secondary school right through to higher education. You need to get clued up on blended learning. Start with Moving to Blended Learning, Part 1: Terminology and Concepts, then take a look the video below with Steve Pettifer explaining techniques for slides that work for blended learning videos:

When you teach, think about how you can support students before and after your time in the classroom.

[1] Conquer the command line is part of the The MagPi essentials series, there are lots of others like it you may find useful on using the camera module, gaming in python, simple electronics and more at https://store.rpipress.cc

Figure 5.4: If you're making presentations for students to watch before or after your lessons, its worth spending some time thinking about how to make your slides video friendly. You can also watch the full 7 minute video embedded in this figure at youtu.be/Y50mTVIzAYk (Pettifer, 2020)

5.2.4 When do the projects start and finish?

Projects start annually in September and are handed at Easter time, see final year project guidelines. For more information contact Duncan Hull.

5.3 Getting a head start

So you've selected *Coding their future* as your third year project and now you are ready to get started. There are plenty of resources to help you prepare and improve your effectiveness in the classroom.

- **Work with young people**
  - Any experience of working with young people will be beneficial, if you have the time and opportunity over the summer you can get some experience of working with young people for example through codeclub.org and coderdojo.com. If you're in Manchester over the summer, see mcrcoderdojo.org.uk which meets every month at the Manchester Technology Centre on Oxford Road, they are always looking for new volunteers.
  - It doesn't have to be coding either, you could get involved with any activities such as sporting events or others, see find-volunteering.manchester.ac.uk
- **Read around** There are lots of really good articles and books to help you, I recommend you start with:
  - *Teaching Computing* (Simmons and Hawkins, 2015) by Carl Simmons & Claire Hawkins, is aimed at teachers of computing, particularly trainee computer science teachers. e-book available here, University of Manchester login required
  - *Computer Science Education* (Sentance et al., 2018) edited by Sue Sentance, Erik Barendsen and Carsten Schulte. Only available in print form, useful for reference
  - *The Cambridge Handbook of Computing Education Research* edited by Sally Fincher and Antony V. Robins (Fincher and Robins, 2019), electronic copies available at doi.org/gb5d (UoM login required) this is a comprehensive overview of the field, useful for reference
  - *Hello World* magazine, see helloworld.raspberrypi.org is packed full of ideas for teaching computing (and its free too)

  - *Quick Reads* give short two page introductions to teaching computing, see blog.teachcomputing.org/tag/quickread
- **Do some courses** there are lots of free Open University courses to get prepared for teaching:
  - Impact of Technology: How To Lead Classroom Discussions. Learn how to keep 14-16 year-old students engaged in discussions while teaching computer science. Supported by Google futurelearn.com/courses/impact-of-technology
  - Teaching Physical Computing with Raspberry Pi and Python futurelearn.com/courses/physical-computing-raspberry-pi-python
  - Since some of your teaching is likely to be asynchronous, you would also benefit from having a look at taking your teaching online from OpenLearn
  - Many more Teaching Computing Courses at futurelearn.com/subjects/teaching-courses/teaching-computing
- **Join the community** The UK Special Interest Group (SIG) on Computer Science Education (CSE) has monthly events listed online at uki-sigcse.acm.org/events, these cover computing events from primary through to secondary and higher education

Chapter 6

Will you be my referee?

As industrial experience and employability tutor, I get asked to write *lots* references for students applying for jobs and further study, see figure 6.1. I'm happy to do this **if** you've been my personal tutee or I've worked with you closely outside of ordinary teaching. However, it's impossible to for me to say YES to every request for a reference. For students I haven't worked with, it is difficult to do, because all I can confirm is facts (attendance, academic marks, degree program) without opinions which doesn't make for a very compelling reference.

Whoever agrees to be your referee, make sure you read and understand the following:

6.1 Who can provide a reference for me?

The best person to provide a reference for you is somebody who knows you, such as your personal tutor. See the careers service guide what are references and how should I choose a referee? and guidance to staff providing references for students from the University of Manchester, which gives extra context.

It is good to have references from different sources, so if you are providing several referees try to pick people from inside and outside the University. Within the University, this is most likely to be your tutor:

- Your personal tutor from year one

Referees may be assisted by umpires, linesmen, timekeepers, or touch judges

Referee

Figure 6.1: I'm not *that* kind of referee, but I *can* provide facts and opinions about you to employers subject to conditions outlined on this page. Whoever your referee is, make sure you ask their permission beforehand, otherwise you may receive an unexpected (virtual) penalty card from your referee. CC BY image of a referee by Jason Gulledge on Wikimedia Commons w.wiki/3dqE adapted using the Wikipedia app

- Your personal tutor from year two (if different to first year)
- Your Industrial Experience (IE) tutor (sometimes that's me)
- Your third year project supervisor
- Your Masters project supervisor
- Anyone else who knows you personally

If you ask somebody who does not know you very well to write a reference for you, all that they are able to do in a reference is confirm rather dull facts such as your grades, your attendance, start date and graduation date. As I've already said, this does not make for a very useful reference.

6.2 Should I ask permission from my referee?

You should always ask the person providing your reference, see figure 6.1.

6.3 What is a reference for?

References have two main purposes:

1. Providing and confirming *facts*
   i. to give a factual account, e.g. of academic record, attendance, etc
   ii. to confirm the accuracy of statements made in an application
2. Providing *opinions*
   i. to give the referee's opinion as to the candidate's suitability for the post/course in question, and his/her potential for the future

6.4 How can I help my referee?

It can make it much easier for your referee if you provide them with information you would like them to mention in your reference. This might include:

- an updated CV

- comments on your character
- any relevant experience you have
- any relevant projects you have completed
- specific aspects of your academic performance

- what the reference is for e.g. a job, Masters or PhD application

All of this information will help your referee to

6.5 Can I have a copy of my reference?

It is unusual for a referee to provide a reference directly to its subject (that's you).

Typically, a referee is asked to provide a reference for a student (or former student) directly by the organisation concerned. For example, if you're applying for postgraduate study, the reference request will be sent by the University directly to your referees email address, who will usually respond by clicking on a link to upload the reference document.

You can, once you're employed, make a request to see your reference under data protection law.

Chapter 7

Contact

You can get in touch using the details below, which include directions and parking information.

7.1 Office

My office is in the Kilburn building. If you exit the building from the Lower First floor, go through the double doors, down the ramp on the South side of the building.

Dr. Duncan Hull, Lecturer

- Room LF25, Kilburn Building
- email: duncan.hull ATE manchester.ac.uk
- telephone: +44 161 275 6186
- linkedin.com/in/duncanhull

Mabel Yau, Careers and placements officer

- Engineering building
- email: mabel.yau ATE manchester.ac.uk
- linkedin.com/in/mabel-yau

Student Support Office

- Engineering building foyer
- email compsci-sso-ug@manchester.ac.uk

Figure 7.1: Paying homage to Alan Turing at a mural on the Princess Parkway by tankpetrol.com. According to Jonathan Swinton, Turing is the "patron saint of Manchester" (Swinton, 2019). As a Manchester icon, he is commemorated locally by the Alan Turing building, the Alan Turing Memorial, some blue plaques and the Alan Turing Way (Cooksey, 2013)

- email compsci-sso-pg@manchester.ac.uk
- telephone: +44 161 549 1288

7.2 Video conferencing

The University of Manchester has an enterprise subscription for Zoom and Microsoft Teams. For the latter, you can contact me using zoom.us/my/duncanhull. For most other enterprise video conferencing software, your employer will need to host the meeting and send me an invitation link. Please provide any log in, dial in, invitation links or meeting IDs for:

- Skype for business
- Bluejeans
- Microsoft Teams
- Cisco Webex

You can also contact me using free versions of:

- Amazon Chime
- Discord
- Slack
- Skype (the non-business version where my username is "duncanhull")
- Jitsi
- Google Hangouts Google id is my University email address

7.3 Online and social media

You can get in touch via t'internet at:

- Blog: duncan.hull.name
- Github: github.com/dullhunk
- Twitter: twitter.com/dullhunk

7.4 Postal address

Send post by snail mail to :

Dr. Duncan Hull
Lecturer
Department of Computer Science
Kilburn Building
The University of Manchester
Oxford Road
Manchester
M13 9PL
Lancashire

7.5 Kilburn building directions

From the train stations, it takes about 20 minutes to walk from Manchester Piccadilly (MAN) and ten minutes from Manchester Oxford Road (MCO). Our official postcode (M13 9PL) takes you to University Place next door, so you're better of using the what3words locations (Leatherdale, 2019) below which are more accurate:

- Google map of the Kilburn building bit.ly/directions-to-kilburn-building
- There are two ground floor entrances to the Kilburn building, North and South
  - North entrance: what3words.com/port.museum.rips
  - South entrance: what3words.com/common.wiping.email
- There is no formal reception so the best place to meet is bit.ly/ByteCafe on the first floor
- See also cs.manchester.ac.uk/about/maps-and-travel/

7.6 Parking

If you are driving, the nearest car parks are:

- **University Car Park B** Manchester Aquatics Centre Car Park, Apcoa M13 9SS
- **University Car Park D** Booth Street West Car Park, M15 6AR, access via Higher Cambridge Street
- See estates.manchester.ac.uk/services/operationalservices/carparking

Appendix A

Computing the future

> One hundred years of computing from 1948 to 2048, stopping at the classes of 2003 and 1978, with some pictures, dad jokes and geek history.

A.1 The class of 2003

Let's start our computational odyssey in 2003, figure A.1 shows students of the MSc Computer Science class of 2003. That's me in the back row standing highest on the left hand side. We are smiling because we were being entertained and educated by Richard Giordano, who took this picture while teaching. We are part of a community of more than 10,000 students have graduated a degree in Computer Science (undergraduate or postgraduate) since the University was the first in the UK to provide an undergraduate Computer Science degree course in 1965. (Brackenbury, 2005)

Twenty five years earlier, before many of the people in the above picture were born, there was the class of 1978…

A.2 The class of 1978

Figure A.2 shows Manchester staff operating the MU5 computer sometime around 1978. Most people will have little interest in the hardware that the people in the picture are using. However, a wider audience may be interested

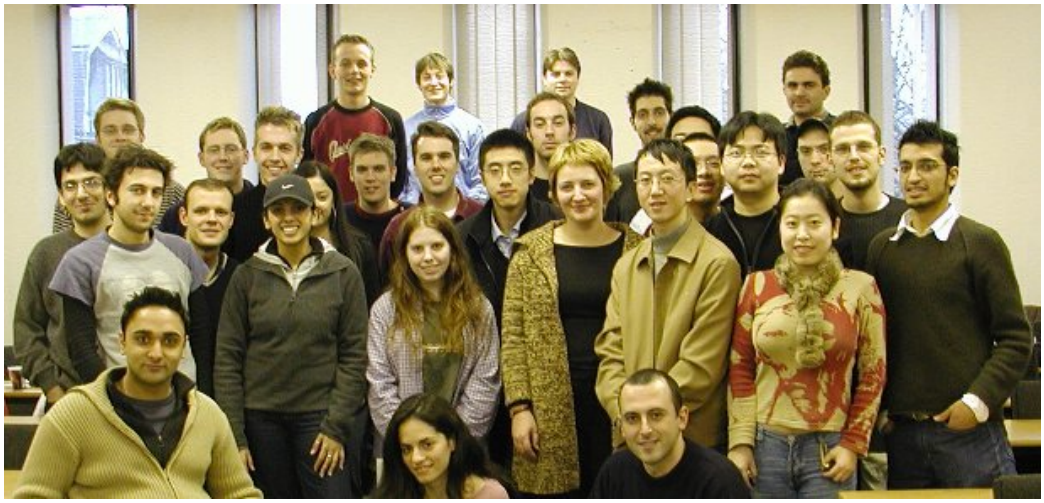

Figure A.1: The class of 2003 including Matthew Horridge, Michael Parkin, Karim Nashar, Iain Hart and myself in the back row

in the suave and sophisticated seventies *Saturday Night Fever* dancing shoes that Pat McKissack on the left of the picture is wearing. (Gibb et al., 1977)

Now, Java junkie James Gosling has fond memories of using a whopping 8K of memory from around about the same period. (Gosling, 2004) But the class of 1948 (see section A.3) were alive today and had seen the Four Yorkshiremen sketch (Brooke-Taylor et al., 1967) they'd probably say:

> Memory? Memory! LUXURY….'Ere in Manchester, when we were lads and built computers, we didn't even have any memory. We had to make our own from an old bit o' tube we found lying around in t'lab. 'Course it only stored 2048 bits and worked for a few hours. Ohhhh we used to DREAM of havin' 8K of memory….

They may have had to use very primitive memory, but at least Tom and Freddie didn't have to build their computers out of mechanical discs and wheels, or even better Meccano. (Hartree and Porter, 1935; Darwin, 1958) Moving on swiftly, we go back a little further in time, thirty years earlier…

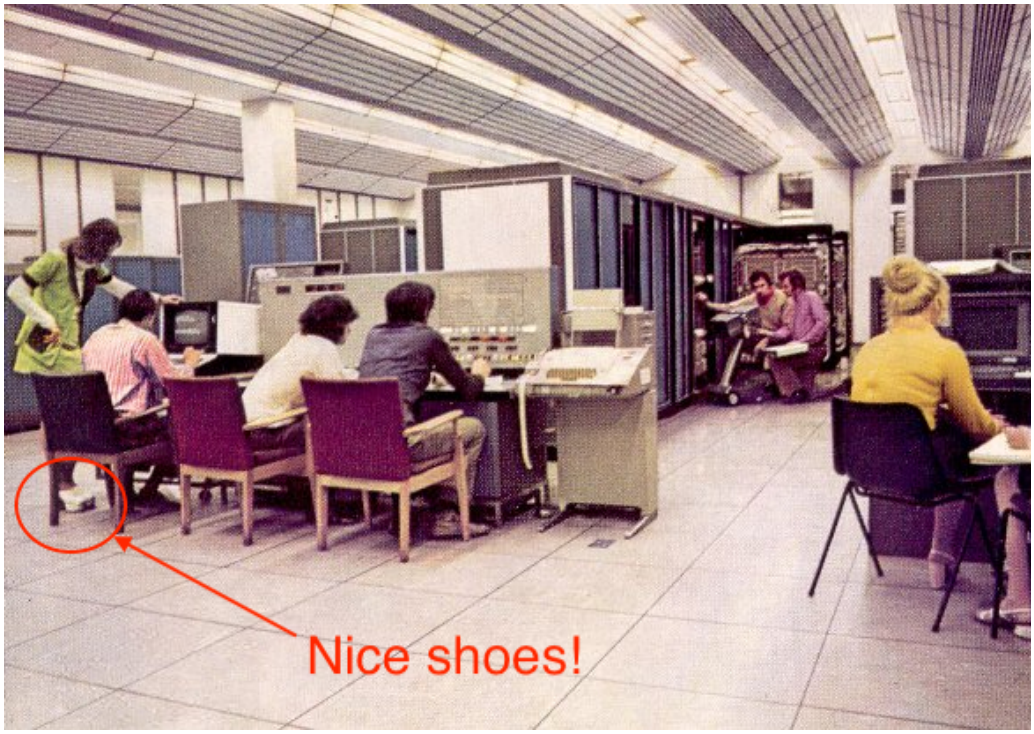

Figure A.2: "You know Simon once we've debugged this code, we're going straight down the discotheque to dance the night away to that groovy new hit record by those local Manchester lads the Bee Gees. How does it go? *Night fever, night fever, we know how to show it.*" The people in the picture are, from left to right, Pat McKissack (wearer of fantastic shoes), Simon Lavington, Gordon Frank, Roland Ibbett, Peter Whitehead, Tony Whitehouse and Lynne Plant (seated at the system performance monitor). The MU5 console is in the foreground with the big machine in the background. The picture is probably taken sometime between about 1972 and 1978, but we've gone for 1978 so that we can make gratuitous references to seventies disco music. See the original unadulterated image via Simon Lavington and the Computer Conservation Society. (Lavington, 2008)

A.3 The class of 1948

Our next stop is in 1948. We have to stop here because the Manchester Baby first ran on 21st June 1948. Figure A.3 shows Tom and Freddie with the Baby. (Lavington, 1998)

Figure A.3: Freddie Williams and his PhD student Tom Kilburn programming the Manchester baby in 1948.

Armed with some double-sided sticky-tape, several empty fairy-liquid bottles, lots of patience and an idea from some bloke called Alan Turing, Tom and Freddie built the world's first stored-program computer. The stored-program was significant as it was the predecessor to what we now call random-access memory (RAM), see figure A.4. (Webb, 2013)

Does this mean the Manchester Baby is the world's first computer? Well, it depends on your definition of computation: It is surprisingly hard to define what counts as a computer and who built the first one (Malcolm, 2000).

Whatever your definition, after Tom and Freddie, things were never the same again. The rest, as they say, is history. That computer you are using…

Figure A.4: This video was produced by Google as a tribute to the Manchester Baby — the first computer to run a program electronically stored in its memory. You can also watch the full 7 minute video embedded in this figure at youtu.be/cozcXiSSkwE (Lavington et al., 2013)

- laptop, desktop
- tablet, phone, satnav
- car, a computer with wheels
- plane, a computer with wings
- games console, wearable tech
- smart card, embedded system etc
- household appliances, washing machine, doorbell etc

… is a direct descendant of the rather strange looking machine in figure A.3.

A.3.1 Numbers speak louder than words

What's interesting about the difference between 1948 and now are the changes in the efficiency, size and speed of computers, shown in table A.1. It's hard to describe in words the difference between 1948 and now, in this case, the numbers speak much louder than words ever could:

Table A.1: Advances in processor power 1948 to 2000, the *Baby in 1948* is the Manchester Baby. The *ARM in 2000* refers the ARM AMULET3H microprocessor taken from CS501: Machine architecture. Thanks to Jim Garside, Doug Edwards and Steve Furber for the data. (Furber et al., 2000; Furber, 2000)

	Baby in 1948	ARM in 2000
Size	Filled a medium sized room	fills 7mm by 3mm of silicon
Power usage (Watts)	3.5kW (3500W)	215mW (0.215W)
Instructions executed (per second)	700	100,000,000
Energy efficiency (Joules per instruction)	5	0.000 000 002 (that's 2,000,000,000 times more efficient than The Baby!)

So where is all this going? What about the future? Let's take a longer view, and skip forward from 1948 to 2048…

A.4 The class of 2048?

Our next stop is 2048. For the number nerds out there, this year is pleasingly 100 years after 1948 so a sensible place to make our next stop. 2048 is also `100000000000` in binary (2^{11}) and the maximum number of **bits** (not `bytes`) that the Manchester Baby stored in its Cathode Ray Tube (CRT) memory. A whopping 2048 bits, or 256 bytes, with 8 bit bytes! (Kilburn, 1948)

What will classes in the year 2048 be studying? Well, in the year 2048, Computer Science won't exist anymore either because:

- As Richard Feynman said Computer Science is not actually a Science (Feynman, 2000)
- As Hal Abelson said the significance of computer science "has little to do with computers" (Abelson et al., 1996)
- As Paul Graham said There is no such thing as "Computer Science" (Graham, 2003, 2004)
- As George Johnson said All Science Is Computer Science (Johnson, 2001)
- As Bill Gates said Computation is Transforming the Sciences (Gates, 2005)

Either way, what is known as Computer Science today will have become so fundamental to many other areas of research, the discipline will naturally become more closely integrated with them. Take Manchester as an example, the hard-sums people will join the mathematics department, the architecture geeks and hardware nerds will join the engineering department, the Computational Biologists will go and join Life Sciences or Medicine, and so on. Of course, I could be very wrong here! As Niels Bohr in figure A.5 might have once said:

Prediction is very difficult, especially if it's about the future. Which is a good point to close this essay on.[1]

[1]Originally written in 2004, for some reason all my quotes were from men. Updated and refreshed in 2021 from the original at https://web.archive.org/web/20180729110412/http://www.cs.man.ac.uk/~hulld/msc2003.html

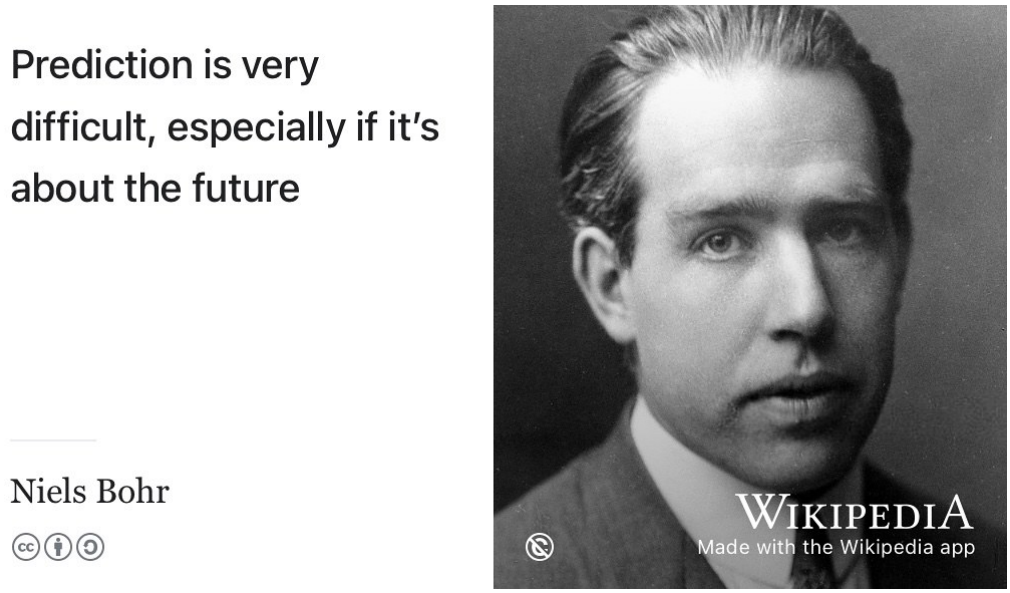

Figure A.5: "Prediction is very difficult, especially if it's about the future". Often attributed to Physicist Niels Bohr, it is disputed if he *actually* said this, see wikiquote.org/wiki/Niels_Bohr#Disputed. Public domain image of Niels Bohr by the Nobel foundation on Wikimedia Commons w.wiki/3dqV

Bibliography

Aaron, S., Blackwell, A. F., and Burnard, P. (2016). The development of sonic pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology and Education*, 9(1):75–94.

Abelson, H., Sussman, G. J., and Sussman, J. (1996). *Structure and Interpretation of Computer Programs*. MIT Press, 2 edition.

Agnew, C. (2020). Widening participation: Annual report.

Anonymous, A. (2013). Why your computer science degree won't get you a job.

Banzi, M. and Shiloh, M. (2015). *Getting Started with Arduino*. O'Reilly Media, Inc, USA.

Bate, A. (2019). Another snazzy raspberry pi wallpaper for your phone and computer. https://www.raspberrypi.org/blog/another-snazzy-raspberry-pi-wallpaper-for-your-phone-and-computer/.

Bell, T. and Vahrenhold, J. (2018). *CS Unplugged—How Is It Used, and Does It Work?*, pages 497–521. Springer International Publishing.

Biggs, J. and Tang, C. (2011). *Teaching for Quality Learning at University*. McGraw-Hill Education Ltd.

Birchall, M. (2019). The times top 100 graduate employers 2019-2020. https://www.top100graduateemployers.com.

Birchall, M. (2020a). Graduate jobs at the UK's top employers grew by 6.2 percent in 2019, the largest annual rise for 5 years. https://www.highfliers.co.uk/download/2020/graduate_market/GM20Release.pdf.

Birchall, M. (2020b). The graduate market in 2020: Annual review of graduate vacancies & starting salaries at the UK's leading employers. https://www.highfliers.co.uk/download/2020/graduate_market/GM20Report.pdf.

Birchall, M. (2021). The graduate market in 2021: Annual review of graduate vacancies & starting salaries at the UK's leading employers. https://www.highfliers.co.uk/download/2021/graduate_market/GM21-Report.pdf.

Birchall, M. (2022). The graduate market in 2022: Annual review of graduate vacancies & starting salaries at the UK's leading employers. https://www.highfliers.co.uk/download/2022/graduate_market/GM22-report.pdf.

Brackenbury, L. (2005). Digital 60: An interview with Linda Brackenbury. https://bbc.in/2Eqyo1K.

Brailsford, D. (2005). Turing complete: computerphile. https://youtu.be/RPQD7-AOjMI.

Briscoe, G. and Mulligan, C. (2014). Digital innovation: The hackathon phenomenon. http://qmro.qmul.ac.uk/xmlui/handle/123456789/11418.

Brooke-Taylor, T., Cleese, J., Chapman, G., and Feldman, M. (1967). Four yorkshireman: Monty python comedy sketch. https://en.wikipedia.org/wiki/Four_Yorkshiremen_sketch.

Cooksey, K. (2013). Alan turing: Manchester celebrates pardoned genius.

Cooper, P. and D'Inverno, R. (2005). The future of the discipline? mathematics and the undergraduate ambassadors scheme. *Journal of Mathematics Teacher Education*, 8(4):329–342.

Darwin, C. G. (1958). Douglas rayner hartree, 1897-1958. *Biographical Memoirs of the Fellows of the Royal Society*, 4:102–116.

Davis, E. (2014a). The case for making hebden bridge the UK's second city. https://bbc.in/3Qst9lF.

Davis, E. (2014b). Mind the gap: London v the rest. https://bbc.in/36KWP6i.

de Vries, R. and Rentfrow, J. (2016). A winning personality: The effects of background on personality and earnings.

DuCharme, B. (2013). *Learning SPARQL*. O'Reilly UK Ltd.

Dylan, B. and Hewson, P. (1988). All along the watchtower. In Iovine, J., editor, *Rattle and Hum*. U2, Island Records. album.

Eilish, B. and O'Connell, F. (2019). Bad guy. In O'Connell, F., editor, *When We All Fall Asleep, Where Do We Go?* Billie Eilish, Interscope Records.

Embury, S. (2022). Comp23311 software engineering 1. https://studentnet.cs.manchester.ac.uk/ugt/COMP23311/syllabus/.

Eraslan, S., Ríos, J. C. C., Kopec-Harding, K., Embury, S. M., Jay, C., Page, C., and Haines, R. (2020). Errors and poor practices of software engineering students in using git. In *Proceedings of the 4th Conference on Computing Education Practice 2020*. Association for Computing Machinery.

Facepalm, O. (2015). Dad dancing, the number one way to embarrass children. *The Daily Telegraph*.

Fauvel, J., Flood, R., and Wilson, R., editors (2006). *Music and Mathematics: From Pythagoras to Fractals*. Oxford University Press.

Feynman, R. P. (2000). *Feynman Lectures on Computation*. Westview Press, 1 edition.

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

Fincher, S., Finlay, J., and Davies, S. (2017). Building a graduate employability community in computing: the GECCO workshops.

Fincher, S. and Petre, M., editors (2004). *Computer Science Education Research*. Taylor & Francis.

Fincher, S. and Robins, A. V., editors (2019). *The Cambridge Handbook of Computing Education Research*. Cambridge University Press.

Fogarty, T. (2015). Hackathons are for beginners. https://medium.com/tfogo/hackathons-are-for-beginners-77a9c9c0e000.

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

Fry, H., Ketteridge, S., and Marshall, S., editors (2014). *A Handbook for Teaching and Learning in Higher Education*. Taylor & Francis Ltd.

Furber, S. (2000). *ARM system-on-chip architecture*. Pearson Education, 2 edition.

Furber, S. (2012). Shutdown or restart? the way forward for computing in UK schools. https://royalsociety.org/topics-policy/projects/computing-in-schools/report/.

Furber, S. (2017). *After the reboot: computing education in UK schools*. Royal Society.

Furber, S., Edwards, D., and Garside, J. (2000). AMULET3: a 100 MIPS asynchronous embedded processor. In *Proceedings 2000 International Conference on Computer Design*. IEEE Computer Society.

Garner, R. (2012). Almost 50 per cent of adults can't do basic maths (that means half).

Gates, B. (2005). The future of computing in the sciences. *microsoft.com*.

Gibb, B., Gibb, R., and Gibb, M. (1977). Night fever. In Oakes, B., editor, *Saturday Night Fever*. Bee Gees, RSO Records.

Gosling, J. (2004). Looking back, looking forward with moore's law. http://web.archive.org/web/20041010131526/http://today.java.net/jag/page6.html#51.

Gowers, T. (2016). Maths isn't the problem - the way it's taught is. https://www.theguardian.com/commentisfree/2016/mar/11/maths-isnt-problem-curriculum-lacking-imagination.

Graham, P. (2003). Hackers and painters. http://www.paulgraham.com/hp.html.

Graham, P. (2004). *Hackers and Painters: Big Ideas from the Computer Age*. O'Reilly.

Green, F. and Kynaston, D. (2019). *Engines of Privilege: Britain's Private School Problem*. Bloomsbury Publishing.

Halfacree, G. (2020). *The Official Raspberry Pi Beginner's Guide*. Raspberry Pi Press, 4 edition.

Harkleroad, L. (2006). *The Math Behind the Music*. Cambridge University Press.

Hartree, D. and Porter, A. (1935). The construction and operation of a model differential analyser. *Memoirs and Proceedings of the Manchester Literary and Philosophical Society*, 79:51–70.

Hawthorne, E. K., Pérez-Quiñones, M. A., Heckman, S., and Zhang, J. (2019). Proceedings of the 50th ACM technical symposium on computer science education. Association for Computing Machinery.

Hodson, R. (2015). Wikipedians reach out to academics. *Nature*.

Howard, H. (1951). Country music hall of fame and museum: Three chords and the truth. https://web.archive.org/web/20200926193611/https://www.countrymusichalloffame.org/content/uploads/2019/05/WM-3-6-Harlan-Howard-Bio.pdf.

Hull, D. (2015). Improving the troubled relationship between scientists and wikipedia. In *First Wikipedia Science Conference, Wellcome Trust, London*. Figshare.

Hull, D. (2017). Wikipedia at the royal society: The good, the bad and the ugly. In *Physiology Friday: The Physiological Society, Hodgkin Huxley House, London*. Figshare.

Hull, D. (2020a). Getting started with computing education projects. http://git.io/computinged.

Hull, D. (2020b). The northern software house: tech employers in the north west #NotJustLondon. https://git.io/manc.

Hull, D. (2022a). *Coding your Future: A guidebook for students*. github.com.

Hull, D. (2022b). Debugging your future. In *Coding Your Future: A Guidebook for Students*. University of Manchester, github.com.

Hull, D. (2022c). Hacking your future. In *Coding Your Future: A Guidebook for Students*. University of Manchester, github.com.

Hull, D., Pettifer, S. R., and Kell, D. B. (2008). Defrosting the digital library: Bibliographic tools for the next generation web. *PLOS Computational Biology*, 4(10):e1000204.

Hull, D., Wolstencroft, K., Stevens, R. D., Goble, C. A., Pocock, M. R., Li, P., and Oinn, T. (2006). Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(Web Server):W729–W732.

Johnson, G. (2001). The world: In silica fertilization; all science is computer science. https://web.archive.org/web/20150527173311/https://www.nytimes.com/2001/03/25/weekinreview/the-world-in-silica-fertilization-all-science-is-computer-science.html.

Kilburn, T. (1948). *A storage system for use with binary digital computing machines*. University of Manchester.

Klassner, F. and Anderson, S. D. (2003). LEGO mindstorms: Not just for k-12 anymore. *IEEE Robotics & Automation Magazine*, 10(2):12–18.

Kubica, J. (2012). *Computational Fairy Tales*. CreateSpace Independent Publishing Platform.

Lavington, S. (1998). *A history of Manchester computers*. British Computer Society, 2nd edition.

Lavington, S. (2008). Where did it all go? In *RESURRECTION: The Bulletin of the Computer Conservation Society*, number 45. Computer Conservation Society.

Lavington, S., Burton, C., Tootill, G., and Edwards, D. (2013). Manchester baby: world's first stored program computer. https://youtu.be/cozcXiSSkwE.

Leatherdale, D. (2019). What3words: The app that can save your life. https://bbc.in/3dp2zuO.

Littlejohn, A., Hood, N., Rehm, M., McGill, L., Rienties, B., and Highton, M. (2019). Learning to become an online editor: the editathon as a learning environment. *Interactive Learning Environments*, pages 1–14.

Malcolm, C. (2000). Who made the first computer?

Mandilaras, A. (2004). Industrial placement and degree performance: Evidence from a british higher institution. *International Review of Economics Education*, 3(1):39–51.

Mohammad-Qureshi, S. and Hull, D. (2019). Learn to edit wikipedia: Thurs-

day 17th october, university of manchester. https://wiki-loves-scientists.org.uk/2019/10/09/ada.

Moller, F. and Powell, S. (2019). Teaching computing via a school placement. In *Proceedings of the 3rd Conference on Computing Education Practice - CEP '19*. ACM Press.

Morris, T. (2019a). Comp30030 third year project laboratory syllabus: Joint honours.

Morris, T. (2019b). Comp30040 third year project laboratory syllabus: Single honours.

Oinn, T., Greenwood, M., Addis, M., Alpdemir, M. N., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M. R., Senger, M., Stevens, R., Wipat, A., and Wroe, C. (2006). Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, 18(10):1067–1100.

Ovenden, M. (2019). Manhattan-chester: Unprecedented residential building in manchester is happening - very high up. nine towers of over 25 storeys have appeared in the past three years, so what's fuelling the change? https://bbc.in/2WVdxzk.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books.

Pettifer, S. (2020). Moving to blended learning part 6: Making video-friendly slides. https://youtu.be/Y50mTVIzAYk.

Potter, J. A., Press, M. C., Callaghan, T. V., and Lee, J. A. (1995). Growth responses of Polytrichum commune and Hylocomium splendens to simulated environmental change in the sub-arctic. *New Phytologist*, 131(4):533–541.

Proffitt, M., editor (2018). *Leveraging Wikipedia: Connecting Communities of Knowledge*. American Library Association.

Rajan, A. (2019). How to break into the elite.

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

Reagle Jr., J. M. (2010). *Good Faith Collaboration: The Culture of Wikipedia*. The MIT Press.

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., and Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11):60.

Rodrigo, O., Nigro, D., Williams, H., and Farro, J. (2021). good4u. In Nigro, D., editor, *Sour*. Olivia Rodrigo, Geffen Records.

Rogers, Y., Paay, J., Brereton, M., Vaisutis, K. L., Marsden, G., and Vetere, F. (2014). Never too old: Engaging retired people inventing the future with makey makey. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press.

Rosenthal, J. (2005). The magical mathematics of music. https://plus.maths.org/content/magical-mathematics-music.

Rothwell, N. (2011a). Manchester 2020: The university of manchester's strategic plan.

Rothwell, N. (2011b). Vision for the university of manchester.

Sattler, U. (2022). Comp10120 first year team project. https://studentnet.cs.manchester.ac.uk/ugt/COMP10120/syllabus/.

Sentance, S. (2018). Recent developments in computer science education research.

Sentance, S., Barendsen, E., and Schulte, C., editors (2018). *Computer Science Education: Perspectives on Teaching and Learning in School*. Bloomsbury Academic.

Sentance, S., Waite, J., Hodges, S., MacLeod, E., and Yeomans, L. (2017). Creating cool stuff: Pupils' experience of the BBC micro:bit. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE '17*. ACM Press.

Shadbolt, N. (2016). An independent review of computer science degree accreditation and graduate employability.

Sharwood, S. (2019). Ibm asks remaining staff to take career advice from

HR-bot. https://www.theregister.co.uk/2017/11/14/ibm_staff_referred_to_cognitive_career_advice_bot_myca/.

Shaw, D. (2012). Makey makey: Improvising tangible and nature-based user interfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI '12*. ACM Press.

Simmons, C. and Hawkins, C., editors (2015). *Teaching Computing*. SAGE, 2 edition.

Singh, S. (2005). The undergraduate ambassador scheme (UAS). https://uas.ac.uk.

Smedley, R. (2019). Conquer the command line: The raspberry pi terminal guide.

Smith, N., Sutcliffe, C., and Sandvik, L. (2014). Code club: bringing programming to UK primary schools through scratch. In *Proceedings of the 45th ACM technical symposium on Computer science education - SIGCSE '14*. ACM Press.

Stacey, A. (2009). How to respond to "i was never much good at maths at school". https://mathoverflow.net/questions/5353/how-to-respond-to-i-was-never-much-good-at-maths-at-school.

Stephenson, C. (2018). Pre-college computer science education: A survey of the field.

Swan, C., Beale, C., Avroutine, I., Hodgson, J., Waller, D., Smith-Nunes, G., Kershaw, J., Saeed, S., Dixon, L., Surrall, A., and Pitt, J. (2013). Cambridge GCSE computing MOOC. https://cambridgegcsecomputing.org.

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

Thiele, I., Swainston, N., Fleming, R. M. T., Hoppe, A., Sahoo, S., Aurich, M. K., Haraldsdottir, H., Mo, M. L., Rolfsson, O., Stobbe, M. D., Thorleifsson, S. G., Agren, R., Bölling, C., Bordel, S., Chavali, A. K., Dobson, P., Dunn, W. B., Endler, L., Hala, D., Hucka, M., Hull, D., Jameson, D., Jamshidi, N., Jonsson, J. J., Juty, N., Keating, S., Nookaew, I., Novère, N. L., Malys, N., Mazein, A., Papin, J. A., Price, N. D., Selkov, E., Sigurdsson, M. I., Simeonidis, E., Sonnenschein, N., Smallbone, K., Sorokin, A., van Beek, J. H. G. M., Weichart, D., Goryanin, I., Nielsen, J., Westerhoff,

H. V., Kell, D. B., Mendes, P., and Palsson, B. Ø. (2013). A community-driven global reconstruction of human metabolism. *Nature Biotechnology*, 31(5):419–425.

Timberlake, J., Martin, M., and Shellback (2016). Can't stop the feeling! In Timberlake, J., Martin, M., and Shellback, editors, *Trolls: Original Motion Picture Soundtrack*. Justin Timberlake, Interscope Records.

Turing, A. M. (1937). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265.

Turki, H., Shafee, T., Taieb, M. A. H., Aouicha, M. B., Vrandečić, D., Das, D., and Hamdi, H. (2019). Wikidata: A large-scale collaborative ontological medical database. *Journal of Biomedical Informatics*, 99.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wainwright, O. (2019). Welcome to manc-hattan: how the city sold its soul for luxury skyscrapers. https://www.theguardian.com/artanddesign/2019/oct/21/welcome-to-manc-hattan-how-the-city-sold-its-soul-for-luxury-skyscrapers.

Wang, Z. and Crawford, I. (2018). Who are gaining the highly paid elite placements in UK higher education? *Studies in Higher Education*, 44(11):1960–1974.

Ward, M. (2016). Coderdojo and libraries: Why your library should get involved and how to start your own. *An Leabharlann : The Irish Library*, 25.

Warner, J. and Guo, P. J. (2017). Hack.edu: Examining how college hackathons are perceived by student attendees and non-attendees. In *Proceedings of the 2017 ACM Conference on International Computing Education Research - ICER '17*. ACM Press.

Webb, L. (2013). "you've come a long way, baby": remembering the world's first stored program computer. https://blog.google/around-the-globe/google-europe/youve-come-long-way-baby-remembering/.

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