

Dr. Duncan Hull

Yours truly, last updated on 19 February, 2026

Contents

Hello, World	7
Full stack teaching	8
What's the Story, Jackanory?	8
Toolbox	9
Get a Life!	9
I DUNCANOME	11
1 Studying Your Future	13
1.1 All Years: Debug Your Future	13
1.2 First Year Students	16
1.3 Second Year Students	16
1.4 Penultimate Year Students	16
1.5 Final Year Students	16
1.6 Masters Students	17
1.7 PhD Students	17
1.8 Extracurricular and Flexible Learning	17
1.9 Previous teaching	18
1.10 Teaching Awards	18

2 Employing Your Future	19
3 Coding Your Future	21
3.1 Reading Your Future	21
3.2 Downloading Your Future	22
4 Coding Their Future	25
4.1 Why Teach Coding?	27
4.2 Participating Schools	27
4.3 Guidance for Teachers	29
4.4 Guidance for Students	32
4.5 Getting a Head Start	35
5 Hearing Your Future	37
5.1 From student to professional	37
5.2 Subscribing to your future	39
6 Refereeing Your Future	41
6.1 “Will you be my referee?”	41
6.2 “Who can provide a reference for me?	43
6.3 “Should I ask permission from my referee?	43
6.4 “What is a reference for?”	44
6.5 “How can I help my referee?”	44
6.6 “Can I have a copy of my reference?”	44
7 Researching the Future	47
7.1 SIGCSE	47
7.2 Industrial Mentoring	49
7.3 Coderdojo & Code Club	49

CONTENTS	5
7.4 Wikipedia	50
7.5 Tuning Complete	51
7.6 Publications	54
A Computing the Future	57
A.1 The Class of 2003	57
A.2 The Class of 1978	58
A.3 The Class of 1948	60
A.4 The Class of 2048?	62
B Contacting the Future	65
B.1 Please Try Reading The Friendly Manual First	65
B.2 Student Support	67
B.3 Office in the Kilburn Building	68
B.4 Postal Address	69
B.5 Kilburn Building Directions	69
B.6 Parking at the Kilburn Building	70
B.7 Duncan's Digital Doppelgängers	71
C Reading the Future	73

Hello, World



Figure 1: Hello, World: my name is Duncan and I'm a Senior Lecturer (Associate Professor) in the Department of Computer Science at the University of Manchester. The left hand side of this picture is what I look like now, the right hand side was taken when I was a longer-haired undergraduate.

Hello and welcome to the Department of Computer Science at the University of Manchester. My name is Duncan and I'm a Senior Lecturer (Associate Professor) and Employability Lead. I teach undergraduate & postgraduate students, supervise tutorials & projects while serving as chair of board of the School of Engineering and a member of University senate. I'm interested in improving teaching, learning and the student experience by:

- Developing and delivering *Coding Your Future*: a course & guidebook for students at cdyf.me
- Improving students communication skills by enabling them to start writing their future, debugging their future and speaking 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 and serving on the board of uki-sigcse.acm.org
- Supporting local schools both as a school governor and with the project Coding Their Future
- Updating students weekly on opportunities with the Wednesday Waggle newsletter at waggle.cs.manchester.ac.uk

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. I serve as a school governor on the governing board of a school in Greater Manchester. As well as teaching in the UK I have also taught in India, Japan and America too.

What's the Story, Jackanory?

So 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

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

Get a Life!

Outside of work I sing a bassy-baritone in a local community show choir, look after bees and enjoy learning greek language and culture, preferably in a or on a Π in Greece. A !

Part I

DUNCANOME

Chapter 1

Studying Your Future

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 hours, on youtube.com/@coding-your-future and in lecture theatres.



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, 2025b) and hacking your future

(Hull, 2025c), especially if you haven't written a CV, Résumé or LinkedIn profile before. Once you've checked your CV against short checklist at cdyf.me/debugging#quick you can get feedback from me by submitting your CV (etc) at bit.ly/debug-my-cv or using the QR code in figure 1.2

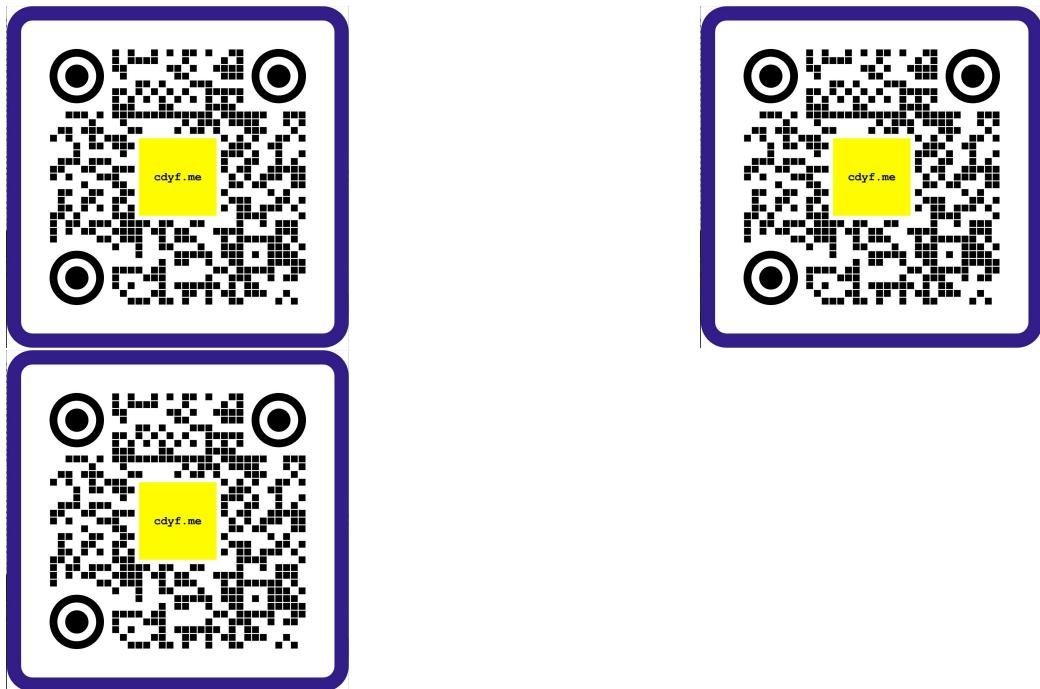


Figure 1.2: Once you've self-checked your CV using cdyf.me/debugging#quick you can get feedback by submitting your CV (etc) at bit.ly/debug-my-cv (login required)

Once you've submitted your CV you can:

- Drop-in to my weekly one-to-one CV clinics for Computer Science students online during term-time during my open office hours, Wednesday 10am to midday on the lower first floor of the Kilburn building
- Get feedback on your CV from as many other people as possible, because “given enough eyeballs, all bugs are shallow” (Raymond, 1999)

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

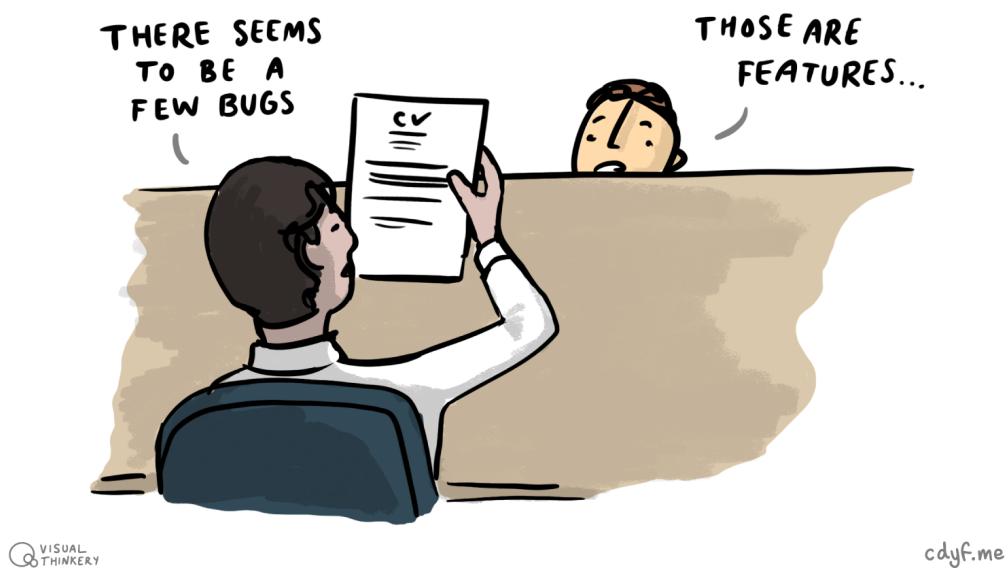


Figure 1.3: Is that a bug or a feature in your CV? It's important to debug your CV before an employer sees it, see the Wednesday Waggle for details of my weekly live debugging sessions. Features not bugs picture by Visual Thinkery is licensed under CC-BY-ND

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 Stewart Blakeway (Blakeway, 2025), 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.

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, 2020)

1.6 Masters Students

If you're doing a Masters degree, you'll find me supervising Master of Science projects in Computer Science and Data Science. (Wickham and Grolemund, 2017) This typically involves various combinations of Wikipedia, Wikidata, 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 and Flexible Learning

Outside all of the above, I'm a promoter and supporter of flexible and extracurricular learning including:

- 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)
- Piloting the introduction of Microsoft Fundamental Certifications across the University in collaboration with the Flexible Learning Program see cdyf.me/achieving#microsoft. (George and Pettifer, 2022)

1.9 Previous teaching

Previously I have served as:

- Course leader for the second year software engineering undergraduate course COMP23311 from 2020 to 2022, which taught students how to make better software
- Organiser of the labs for the software engineering mentoring program
- Teaching team member for second year software engineering: COMP23311 Led by Suzanne Embury and subsequently Bijan Parsia
- Second year tutor from 2016 through to 2020
- Leading the course on Principles of Digital Biology which covers Bioinformatics, Computational Biology and a bit of medical informatics

1.10 Teaching Awards

Thanks to all the students at the University of Manchester Students' Union (UMSU) who nominated me for an academic advising champion award in 2024, see manchesterstudentsunion.com/awards and figure 1.10. These annual awards recognise the achievements of staff and students who have gone above and beyond to represent student interests. (McMahon, 2024)

Thanks to all the students at the UMSU who nominated me for academic advising award and thanks Sambbhav for recording it. UMSU is the largest students' union in the UK representing over 40,000 students. You can also watch the full 30 second video embedded in this figure at youtu.be/pe-M8_lKUz0 (Jackson and Ansari, 2024)

Chapter 2

Employing Your Future

If you are an employer, graduates and undergraduates are an important part of the future of your organisation. Collaborating with us in the Department of Computer Science is one of the best ways to encourage students from the University of Manchester to join your organisation as employees both before and after they graduate, see figure 2.1.

If you're not recruiting graduates, your organisation may suffer in the longer term as talented and young employees decide to work for other employers. Recruiting graduates is a cost effective way to build the skills and knowledge your organisation needs to succeed in the future. Find out more in our industry club pages at industry-club.cs.manchester.ac.uk



Figure 2.1: Are these graduates the future of your organisation? If you want them to be, you need to engage early and often with students *before* they graduate. This will increase your chances of recruiting them. University of Manchester graduates celebrating their graduation outside the Samuel Alexander building with photobombing by Gavin Brown

Chapter 3

Coding Your Future

Your future is bright, your future needs coding. So let's start coding your future.

Coding Your Future is a guidebook written to help students design, build, test and code their futures in computing, see figure 3.1. Written by yours truly, (Hull, 2025a) 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 3.2.

3.1 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

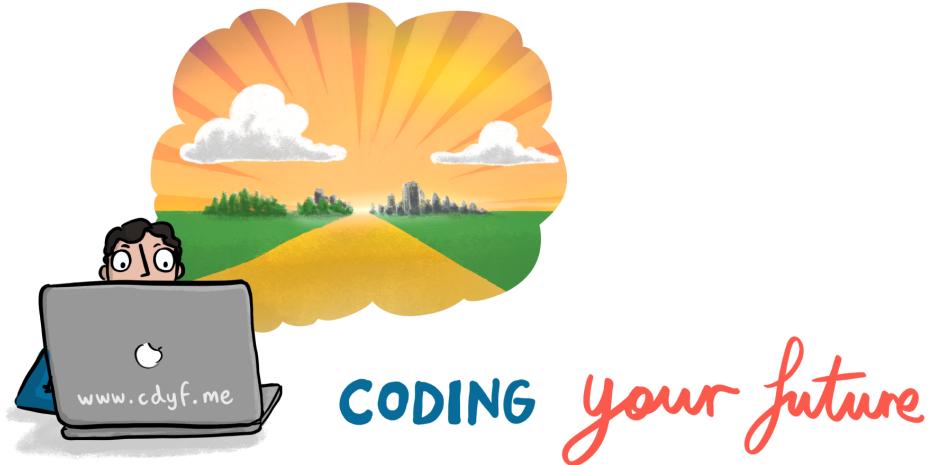


Figure 3.1: Coding your Future is a guidebook to help students design, build, test and code their futures in computing, see www.cdyf.me. *Coding your Future* illustration by Visual Thinkery is licensed under CC-BY-ND

- 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.

3.2 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 4

Coding Their Future

Their future is bright, their future needs coding. So let's start 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 computing education before, during and after GCSE Computer Science, particularly at key stages 3, 4 and 5. (Furber, 2012, 2017; Sentance et al., 2023; Swan et al., 2013)
- widen participation in higher education, especially in under-represented groups. (Palmer, 2023; Friedman and Laurison, 2020; Green and Ky-naston, 2019; Rajan, 2019)
- enable our undergraduate students to develop their leadership and communication skills by working with young people in the classroom

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,

**the art of teaching and
learning the discipline of
computer science**

Computer
science education

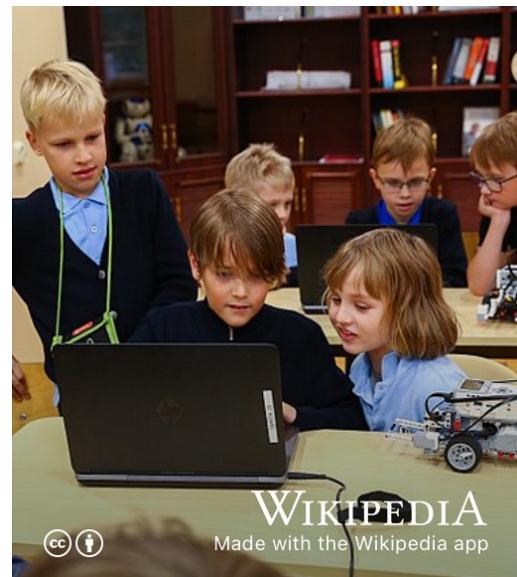


Figure 4.1: Is it art or is it science? It's both: Undergraduate students regularly practice the art of computer science education in secondary schools as part of our wider social responsibility activities (Rothwell, 2024, 2011) and schools, colleges and public activities Picture of children coding adapted from an original by Arno Mikkor on Wikimedia Commons w.wiki/65A2 using the Wikipedia app

2019) except students work is assessed using our final year project framework. (Morris, 2019a,b)

4.1 Why Teach Coding?

Coding, programming, software engineering and computer science - whatever you want to call it, builds key foundational skills and knowledge that **all** students need to learn regardless of their aspirations. Children don't learn to code just to become coders, they learn to code so they can build problem solving skills and develop an understanding how the digital world they inhabit functions. (Jones, 2023; Colligan et al., 2025), creator of the sonic-pi.net makes this case in figure 4.1.

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/0lTZ8Tuyu5I (Aaron, 2016)

Like Maths and English: Coding is for everybody, not just the techies.

4.2 Participating Schools

Since these projects were started in 2012, our undergraduate students have worked in twenty different schools in Greater Manchester. Participating organisations are mostly non-selective state schools, to align with the University of Manchester's values of social responsibility and civic engagement. Current and previous participating schools are shown in table 4.1, ranked by the percentage of their students entitled to free school meals (FSM).

Table 4.1: Schools our undergraduate students have worked with in teaching computing in since 2013, ordered by the percentage of students entitled to free school meals (FSM). As of 2023, the UK average for students entitled to FSM according to gov.uk is **23.8%**. (Servant, 2023) This number has been increasing over the last decade as the UK falls behind on tackling child poverty (Foster and Borrett, 2024; Editor, 2022) Note that all private schools have an FSM percentage of zero because their students do not qualify for benefits, which makes it difficult to compare intakes of state and private schools. It seems likely that children in private schools probably don't need free school meals anyway.

School	Location	% FSM via gov.uk
theeastmanchesteracademy.co.uk	Beswick	65.9%
manchestercommunicationacademy.com	Harrow	54.9%
wrhs1118.co.uk	Whalley	41.6%
	Range	
stpetershigh.com	Longsight	41.4%
allhallowssalford.com	Salford	37.4%
thebarlowrchigh.co.uk	Didsbury	37.2%
didsburyhighschool.org.uk	Didsbury	35.6%
stannes.academy	Stockport	34%
pwhs.co.uk	Didsbury	31.3%
utcmediacityuk.org.uk	Salford	31.3%
trinityhigh.com	Hulme	31.1%
whgs-academy.org	Whalley	30.9%
	Range	
“Bog Standard Comprehensive”	Anywhere	24.6%
fairfieldhigh.tameside.sch.uk	Droylsden	23.1%
laurusheadlehulme.org.uk	Cheadle	16.6%
	Hulme	
knutsfordacademy.org.uk	Knutsford	12.4%
chhs.org.uk	Cheadle	11.1%
	Hulme	
blue-coat.org	Oldham	10.3%

School	Location % FSM via gov.uk
stretfordgrammar.com	Stretford 9.9%
agsb.co.uk	Altringham 3%
aggs.bright-futures.co.uk	Altringham 26%
mgs.org	Rusholme 0%

The project was setup by Duncan Hull and David Rydeheard and is now led by Duncan. There are lots of open questions about how Computer Science should be taught. (Sentance et al., 2023; Sentance, 2018; Stephenson, 2018; Fincher and Petre, 2004) To find out more, see the guidance for teachers and guidance for students below.

There's a well documented shortage of Computer Science teachers in the UK.(Common and Pledger, 2023; Roberts, 2022; Editor, 2019) Students working in schools can address this issue by learning from and supporting current teachers of Comptuer Science.

4.3 Guidance for Teachers



Figure 4.2: 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))

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.

4.3.1 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

4.3.2 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.

4.3.3 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.

4.3.4 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

4.3.5 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 be beneficial for both your school and the University. We don't want to burden you with unnecessary bureaucracy that all teachers battle with!

4.4 Guidance for Students



Figure 4.3: 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, 2020) 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

4.4.1 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.

4.4.2 What will 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
- a substantial piece of software or hardware, this is a requirement for all projects in Computer Science

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,

¹Conquer 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>

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.

4.4.3 Blended Learning

COVID 19 has accelerated changes in the way we teach and learn, from primary and secondary school right through to higher education. As a teacher, you need to get clued up on blended learning. A good place to start is 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:

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)

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

4.4.4 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.

4.5 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 mrccoderdojo.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., 2023) 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 uk-sigcse.acm.org/events, these cover computing events from primary through to secondary and higher education

Chapter 5

Hearing Your Future

Hearing Your Future is a podcast which accompanies the Coding Your Future guidebook. We talk to graduands and graduates of computer science, see figure 5.1, to hear more about their journey from student to professional. As part of their exit interview we ask them what happens after graduation? What comes next?

5.1 From student to professional

Listen in to the episodes below, or read the transcripts, to find out more about how these students are coding their future:

- What's their story?
- How did they get to where they are?
- What obstacles have they faced and how did they overcome them?
- What advice would they offer to their former selves and fellow students?
- Where are they planning to go next?

Find out more by listening here or subscribing wherever you get your podcasts.

Your future is bright, your future needs hearing, so let's start hearing your future.



Figure 5.1: Some recent guests on the podcast. Collage of pictures re-used from LinkedIn and Github with permission. From left to right, top to bottom: Steve, Cristian, Stefania, Bozhidar, Eman, Asma, Jonathan, Ivo, Ingry, Nadine, Pedro, Amish, Raluca, Jason, Brian, Carmen, Sneha and Alice. Thanks again to all our guests for telling us your stories on *Hearing Your Future*, the Coding Your Future podcast.

5.2 Subscribing to your future

You can subscribe (for free!) by following one of the links below to your favourite podcast tool. Alternatively you can search for **Coding Your Future or Hearing Your Future** in your podcast application. You will see the logo shown in figure 5.2 when you find it.

- Apple: <apple.co/3JFV5Qe>
- Spotify: <spoti.fi/3SFqaYk>
- Amazon: <amzn.to/3OOaS0G>
- Google: <music.youtube.com/podcasts> (appearing shortly)
- Deezer: <deezer.com/en/show/3839607>
- Libsyn: <codingyourfuture.libsyn.com>

If all else fails, you can cut-and-paste the raw **rss** feed URL below into the relevant settings on your podcast app:

```
http://feeds.libsyn.com/299189/rss
```



Figure 5.2: *Hearing your future* is the Coding your Future podcast. Subscribe and listen wherever you get your podcasts, see section 5.2. Hearing sketch by Visual Thinkery is licensed under CC-BY-ND

Chapter 6

Refereeing Your Future

So, you're a student who needs a reference for a new job or future study?

6.1 “Will you be my referee?”

This is a question academics get asked a lot. As industrial experience and employability tutor, I get asked that question more than most and already write *lots* of references for students applying for jobs and further study, see figure 6.1.

Unfortunately, like yours, my time is a precious and finite resource. So I can write a reference for you if:

- You have been my personal tutee in first, second, placement or final year
- We have worked together outside of ordinary teaching

It is impossible for me (or any other academic) to say YES to every request for a reference. It is time consuming and difficult for academics to write lots of references for students they don't know much about, as all they can do is confirm facts (attendance, academic marks, degree program) without opinions. This doesn't make for a very compelling reference and can be provided by administrators at the University anyway.

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

Referee



Figure 6.1: Come on ref! 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 Gulleedge on Wikimedia Commons w.wiki/3dqE adapted using the Wikipedia app

You can obtain an unofficial copy of your transcript by emailing soe.hub@manchester.ac.uk.

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

6.2 "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
- 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.

6.3 "Should I ask permission from my referee?"

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

6.4 “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

6.5 “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 write a better reference for you.

6.6 “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

Researching the Future

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. Previously I've worked in bioinformatics, knowledge representation and reasoning and symbolic AI, see publications in section 7.6.

7.1 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 its Special Interest Group (SIG) in Computer Science Education (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

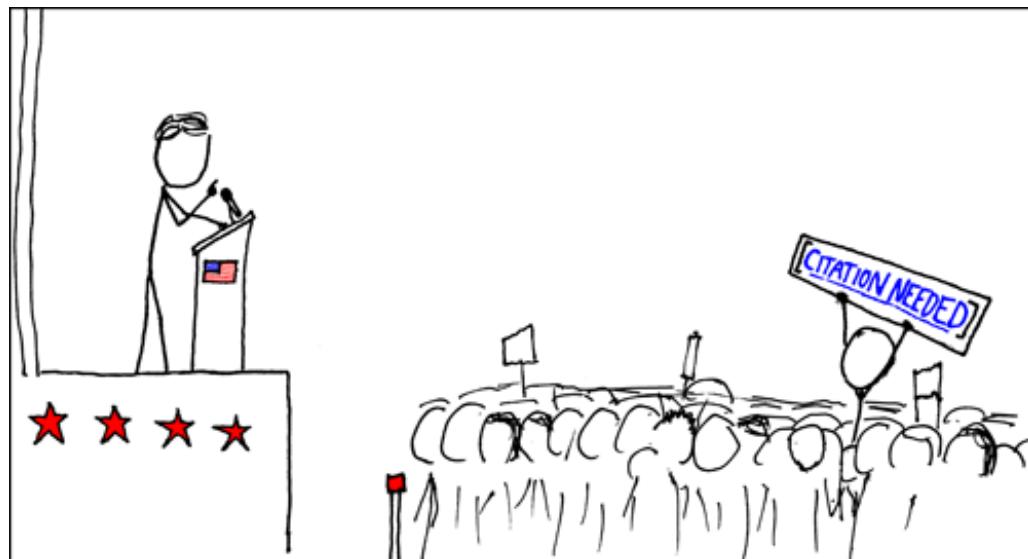


Figure 7.1: Too many educational practices are not backed up by good evidence that they actually work. More evidence is needed to support many of the claims made about effective pedagogy. *Wikipedian Protester* cartoon by Randall Munroe at xkcd.com/285 published under a Creative Commons Attribution-NonCommercial 2.5 License

of UK ACM SIGCSE and have served on the program committee for Computing Education & Practice (CEP) conference at Durham University

7.2 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.

7.3 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, (Hal-facree, 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.

7.4 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. 2025-02-12 Raising Representation CAHAE event, run in collaboration with the Department of Classics, Ancient History, Archaeology and Egyptology (CAHAE)
2. 2024-11-07 Raising Representation of Women in Archaeology, run in collaboration with Hannah Cobb
3. 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.
4. 2020-02-26 Wikimedia in Education UK Summit, Coventry University #wikiedu20
5. 2019-11-22 Training of Trainers (ToT) workshop, University of Glasgow

6. 2019-10-19 Learn to edit Wikipedia with Ada Lovelace, Sackville Street Building, University of Manchester (Mohammad-Qureshi and Hull, 2019)
7. 2019-10-12 Wikipedia Edit-a-Thon with Zebra Hub HQ, Pankhurst Centre, Manchester
8. 2017-10-13 Physiology Friday, Hodgkin Huxley House, Farringdon, London (Hull, 2017)
9. 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

7.5 Tuning Complete

Tuning complete started as a boy band in 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 founding members, shown in figure 7.2 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)
- Duncan Hull: MC,
 - synth (MicroKORG)
 - drum machine (Sonic Pi)
 - embarrassing dad dancing (Facepalm, 2015)



Figure 7.2: The founder members of Tuning Complete were Jez Lloyd, Steve Furber, Justin Timberfake and yours truly. With apologies to Richard Avedon and his psychedelic portraits of John, Paul, Ringo and George. Artwork by Jez Lloyd.

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)

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 7.3.



Figure 7.3: 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!

7.5.1 From boy band to Musical Collective

After several years of rehearsing, touring and time in the studio, we decided to invite women to join our boy band, because we soon realised they could

do all the things we weren't so good at such as:

- Singing
- Dancing
- Performing
- Looking good
- Reading and writing music
- Playing musical instruments

So, as of 2025, Tuning Complete are a musical collective with a fluid membership model, we're always looking for new members to join our ensemble. Artists we have collaborated with include:

- Billie Fakish: guest vocalist (Eilish and O'Connell, 2019)
- Olivia Fakerigo: guest vocalist (Rodrigo et al., 2021)
- Faky Perry: guest vocalist (Perry et al., 2013)
- Faker Swift: guest vocalist (Swift, 2019)
- Aretha Fakelin: guest vocalist (Covay and Franklin, 1967)

If you can sing, dance or play a musical instrument and would like to join us, please get in touch.

7.6 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
- research.manchester.ac.uk/en/persons/duncan.hull

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!

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 proud members of a community of more than 10,000 students have graduated with a degree¹ in Computer Science 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...

¹either undergraduate or postgraduate

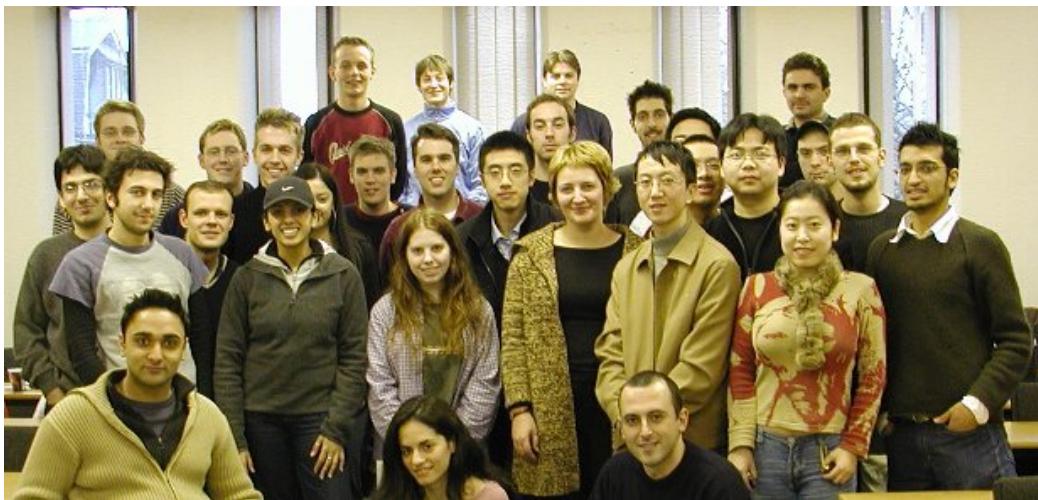


Figure A.1: The class of 2003 including Matthew Horridge, Michael Parkin, Karim Nashar, Iain Hart and myself in the back row

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 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 if 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



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)

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...

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.3. (Webb, 2013)

This video was produced by Google as a tribute to the Manchester Baby — the first computer to run a program electronically stored in its mem-

ory. You can also watch the full 7 minute video embedded in this figure at youtu.be/cozcXiSSkwE (Lavington et al., 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...

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

... is a direct descendant of the rather strange looking machine in figure A.3. In 2023, the BBC published archive footage of Tom Kilburn being interviewed by Brian Redhead in 1981 about the big machine and its successors, shown in figure A.3.

Broadcaster Brian Redhead interviews proud Yorkshireman and Manchester United fan Tom Kilburn in 1981. You can also watch this 14 minute video at youtu.be/CQ2EBw7AgZY (Redhead, 1981) This clip looks *very* dated now (male, pale and stale) but provides an interesting window on the past. It was acceptable in the 80s, it was acceptable at the time. (Harris, 2007)

You don't need to be a Manchester United supporter or from Yorkshire, like Kilburn was, to appreciate computing. A little basic arithmetic helps illustrate how far we've come since 1948 and 1981.

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.4 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.²

²Originally written in ~2006, for some reason all my quotes were from men. Updated and refreshed in 2021 from the original at <https://web.archive.org/web/20060304101200/http://www.cs.man.ac.uk/~hulld/msc2003.html>

Prediction is very
difficult, especially if it's
about the future

Niels Bohr

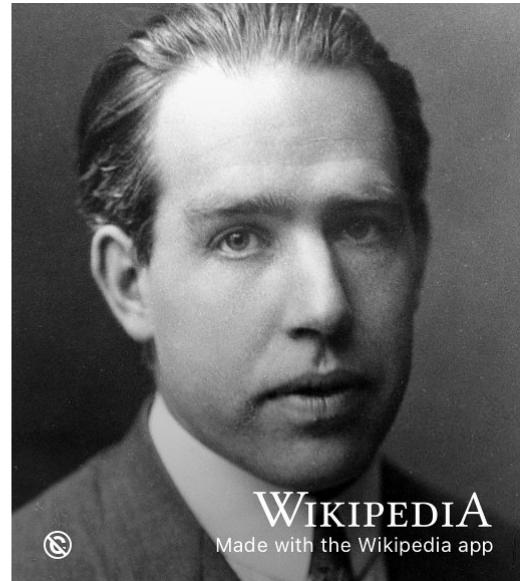


Figure A.4: “Prediction is very difficult, especially if it’s about the future”. Often attributed to Physicist Niels Bohr, it is disputed if he *actually* said this, see [wikiquote.org/wiki/Niels_Bohr#Disputed](https://www.wikiquote.org/wiki/Niels_Bohr#Disputed). Public domain image of Niels Bohr by the Nobel foundation on Wikimedia Commons w.wiki/3dqV

Appendix B

Contacting the Future

The future has arrived — It's Just Not Evenly Distributed Yet.
(Gibson, 1992)

Unfortunately you can't contact the future, yet. While we wait for the future to become more evenly distributed you *can* contact me using the information below which includes directions to the Kilburn Building and parking information.

B.1 Please Try Reading The Friendly Manual First

If you're a student, please make sure you've Read The Friendly Manual (RTFM), before asking me or my colleagues. There's a good chance your question will already be answered there, in particular:

- The Undergraduate (UG) Virtual Common Room bit.ly/UG-VC-2025-26 and links it contains, in particular:
 - The Undergraduate (UG) Handbook bit.ly/UG-handbook-2025-26
 - The Placement Handbook bit.ly/placement-handbook-2025
 - The Monday Mail studentnet.cs.manchester.ac.uk/ugt/mondaymail



Figure B.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)

- The Wednesday Waggle (for jobs) waggle.cs.manchester.ac.uk
- The student support website at www.studentsupport.manchester.ac.uk

B.2 Student Support

If you can't find an answer to your question in section B.1, then get in touch. Bear in mind, I might not be the best person to talk to, depending on what you want. If you're a student, you might be better off contacting student support in person by the neon signs in the Nancy Rothwell Building, see figure B.2.



Figure B.2: The Student Support Office (SSO) on the first floor of the Nancy Rothwell Building are here to help you, they should be your first port of call. If you have something more specific about a personal, academic or employment issue I may be able to help you.

You can contact student support using the emails below:

Query	contact
General UG & PGT Queries	soe.programmes@manchester.ac.uk
Appeals and Conduct	soe.appealsandconduct@manchester.ac.uk
Assessment and Progression	soe.assessment@manchester.ac.uk
Attendance Queries and Issues	soe.attendance@manchester.ac.uk
Programme Changes	soe.programmes@manchester.ac.uk
Course Unit Changes	soe.courseunits@manchester.ac.uk
Mitigating Circumstances, support and Welfare	soe.wellbeing@manchester.ac.uk
Placements	soe.placements@manchester.ac.uk

Query	contact
Timetabling	soe.timetabling@manchester.ac.uk
Student Data	soe.studentdata@manchester.ac.uk
Student Support Hub	soe.hub@manchester.ac.uk
Student Experience ¹	soe.studentexperience@manchester.ac.uk
Visas	visas@manchester.ac.uk
Any Other Business (AOB)	soe.programmes@manchester.ac.uk

If you provide the following information in your email, they will be able to respond to your query more efficiently:

- Name of Department e.g. Computer Science
- Student ID Number
- Programme of Study e.g. BSc Computer Science
- Brief description of how we can assist you

You can do this via bit.ly/student-services-enquiry-form

B.3 Office in the Kilburn Building

My office is on the Lower First floor, Room LF25 of the Kilburn building shown in figure B.3. From the Byte cafe on the lower first floor, go through the double doors on the South side of the building and down the ramp. LF25 is at the bottom of the ramp on the right, see bit.ly/Kilburn-lower-first-floor.

- Room LF25, Kilburn Building
- telephone: +44 161 275 6186

Details of my office hours can be found in the Wednesday Waggle.

¹for FSE careers bulletin



Figure B.3: A panoramic picture of the Kilburn building which looks like a giant brick heat sink. This picture is taken from the Oxford Road showing University Place on the right, which looks like a giant tin of baked beans. Creative Commons BY Image by Yours Truly via Wikimedia Commons w.wiki/E6Qk

B.4 Postal Address

Send post by carrier pigeon or snail mail to:

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

B.5 Kilburn Building Directions

The Kilburn building shown in figure B.3 is about 20 minutes walk from either Manchester Piccadilly (MAN) or ten minutes from Manchester Oxford Road (MCO). Our official postcode (M13 9PL) takes you to University Place (which looks like a giant tin of baked beans) next door, so you're better off using the what3words locations (Leatherdale, 2019) below which are more



Figure B.4: A panoramic picture of the South entrance of the Kilburn building on Wilton Street with the Oxford Road on the left and the Alan Turing Building on the right hand side of the picture.

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, see figure B.5
 - South entrance: what3words.com/common.wiping.email, see figure B.4
- 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

B.6 Parking at the Kilburn Building

If you are driving, the nearest car parks are:

- **University Car Park B** Manchester Aquatics Centre Car Park, Ap-coa M13 9SS



Figure B.5: A panoramic picture of the North entrance of the Kilburn building with the Oxford Road on the right hand side of the picture.

- **University Car Park D** Booth Street West Car Park, M15 6AR, access via Higher Cambridge Street
- See estates.manchester.ac.uk/services/operationalservices/carparking

B.7 Duncan's Digital Doppelgängers

You can connect online with my digital doppelgängers on t'internet at:

- Blog: duncan.hull.name
- Email: firstname.surname@manchester.ac.uk
- Github: github.com/dullhunk
- LinkedIn: linkedin.com/in/duncanhull
- Microsoft Teams: using email firstname.surname@manchester.ac.uk
- YouTube: youtube.com/@coding-your-future
- Zoom: zoom.us/my/duncanhull

Besides Zoom/Teams, I can also use most other video conferencing software including²: Amazon Chime, Bluejeans, Cisco Webex, Discord, Google Hangouts, Jitsi and Slack. (Klein, 2023)

²so much software to choose from!



Figure B.6: Posing in the Googleplex at 1600 Amphitheatre Parkway in Mountain View, California. Thanks to Andrew Lang for taking this picture during Science Foo Camp back in the more optimistic days of “Don’t be evil” (Brin et al., 2004; Hull, 2007, 2009), old skool Google logos (Kedar, 2023) and the unfortunate enshittification of the internet. (Doctorow, 2025, 2024)

Appendix C

Reading the Future

Here's a list of everything cited in these pages.



Figure C.1: How much of the future has already been written?

Bibliography

Aaron, S. (2016). Programming as performance. <https://youtu.be/0lTZ8Tuyu5I>.

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.

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.

Blakeway, S. (2025). Comp10120 first year team project. <https://studentnet.cs.manchester.ac.uk/ugt/COMP10120/syllabus/>.

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>.
- Brin, S., Page, L., and Buchheit, P. (2004). Google code of conduct. <https://web.archive.org/web/20041009211006/http://investor.google.com/conduct.html>.
- 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.
- Colligan, P., Griffiths, M., and Cucuiat, V. (2025). Why kids still need to learn to code in the age of ai: A position paper from the raspberry pi foundation. <https://www.raspberrypi.org/blog/why-kids-still-need-to-learn-to-code-in-the-age-of-ai/>.
- Common, J. and Pledger, D. (2023). Solving the mystery of the missing computer science specialists. <https://www.stem.org.uk/all-news/solving-the-mystery-of-the-missing-computer-science-specialists>.
- 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.
- Covay, D. and Franklin, A. (1967). Chain of fools. In Wexler, J., editor, *Lady Soul*. Aretha Franklin, Atlantic Records.
- Darwin, C. G. (1958). Douglas Rayner Hartree, 1897-1958. *Biographical Memoirs of the Fellows of the Royal Society*, 4:102–116.
- Doctorow, C. (2024). ‘enshittification’ is coming for absolutely everything: The term describes the slow decay of online platforms such as facebook. but what if we’ve entered the ‘enshittocene’. <https://on.ft.com/40ViYOr>.
- Doctorow, C. (2025). *Enshittification: Why Everything Suddenly Got Worse and What to Do About It*. Macmillan Publishers.

- 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.
- Editor, A. (2019). Policy briefing on teachers of computing: Recruitment, retention and development. <https://royalsociety.org/-/media/policy/Publications/2019/21-08-19-policy-briefing-on-teachers-of-computing.pdf>.
- Editor, A. (2022). Free school meals: How many children can claim them? <https://www.bbc.co.uk/news/54693906>.
- 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.
- Facepalm, O. (2015). Dad dancing, the number one way to embarrass children.
- 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 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>.
- Foster, P. and Borrett, A. (2024). How the uk is falling behind on tackling child poverty: Almost a third of children in the country are living in relative poverty, which has increased more than any advanced economy. <https://on.ft.com/3wTvW26>.
- 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. <https://web.archive.org/web/20080221053710/https://www.microsoft.com/presspass/exec/billg/speeches/2005/11-15SuperComputing05.aspx>.
- George, D. and Pettifer, S. (2022). Flexible learning: Our strategy for the future. <https://documents.manchester.ac.uk/display.aspx?DocID=64389>.
- Gibb, B., Gibb, R., and Gibb, M. (1977). Night fever. In Oakes, B., editor, *Saturday Night Fever*. Bee Gees, RSO Records.
- Gibson, W. (1992). The future has arrived — it's just not evenly distributed yet. <https://quoteinvestigator.com/2012/01/24/future-has-arrived/>.
- 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/mathsisnt-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.
- Harris, C. (2007). Acceptable in the 80s. In Wexler, J., editor, *I Created Disco*. Calvin Harris, Columbia Records.
- 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. (2007). Scifoo day 1: Turn up, tune in, drop out. <https://doi.org/10.59350/dzkkf-vw193>.
- Hull, D. (2009). California googlin'. <https://doi.org/10.59350/n8s23-9ys33>.
- 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. (2020). Getting started with computing education projects. <http://git.io/computinged>.
- Hull, D. (2025a). *Coding your Future: A guidebook for students*. github.com.
- Hull, D. (2025b). Debugging your future. In *Coding Your Future: A Guidebook for Students*. University of Manchester, github.com.
- Hull, D. (2025c). 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.
- Jackson, K. and Ansari, R. (2024). University of manchester student's union awards: Academic advising champions 2024. https://www.youtube.com/watch?v=pe-M8_lKUz0.
- 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>.
- Jones, S. P. (2023). Why we must (still) teach our children to code. <https://wwwcomputingatschool.org.uk/news-and-blogs/2023/november/why-we-must-still-teach-our-children-to-code>.
- Kedar, R. (2023). Meet the google logo designer: An interview with ruth kedar, who designed the google logo displayed on the main search page for 16 years. https://store.google.com/intl/en_uk/ideas/articles/google-logo-design/.

- 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.
- Klein, N. (2023). *Doppelganger: A Trip Into the Mirror World*. Farrar, Straus and Giroux.
- 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?
- McMahon, A. (2024). Fse recognised at 2024 su awards. <https://www.teachingcollege.fse.manchester.ac.uk/fse-recognised-at-2024-su-awards/>.
- Mohammad-Qureshi, S. and Hull, D. (2019). Learn to edit wikipedia: Thursday 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.
- Palmer, S. (2023). Access and Student Success Impact Report. <https://documents.manchester.ac.uk/display.aspx?DocID=72419>.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books.
- Perry, K., Gottwald, L., Martin, M., McKee, B., and Walter, H. (2013). Roar. In Cirkut, editor, *Prism*. Katy Perry, Capitol Records.
- 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.
- Redhead, B. (1981). Bbc archive: Tom kilburn - yorkshire's computing pioneer. <https://youtu.be/CQ2EBw7AgZY>.

- 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.
- Roberts, J. (2022). Ofsted warning over shortage of computing teachers. <https://www.tes.com/magazine/news/general/ofsted-warning-over-shortage-computing-teachers>.
- 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. (2011). Manchester 2020: The university of manchester's strategic plan.
- Rothwell, N. (2024). Vision for the university of manchester.
- Sentance, S. (2018). Recent developments in computer science education research.
- Sentance, S., Barendsen, E., Howard, N. R., and Schulte, C., editors (2023). *Computer Science Education: Perspectives on Teaching and Learning in School*. Bloomsbury Academic, 2 edition.
- 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.
- Servant, A. (2023). Schools, pupils and their characteristics: Gov.uk. <https://explore-education-statistics.service.gov.uk/find-statistics/school-pupils-and-their-characteristics>.

- 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. <https://magpi.raspberrypi.com/books/command-line-second-edition/>.
- 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>.
- Swift, T. (2019). You need to calm down. In Little, J., editor, *Lover*. Taylor Swift, Republic Records.
- 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., Weichert, 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.
- 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., 1 edition.