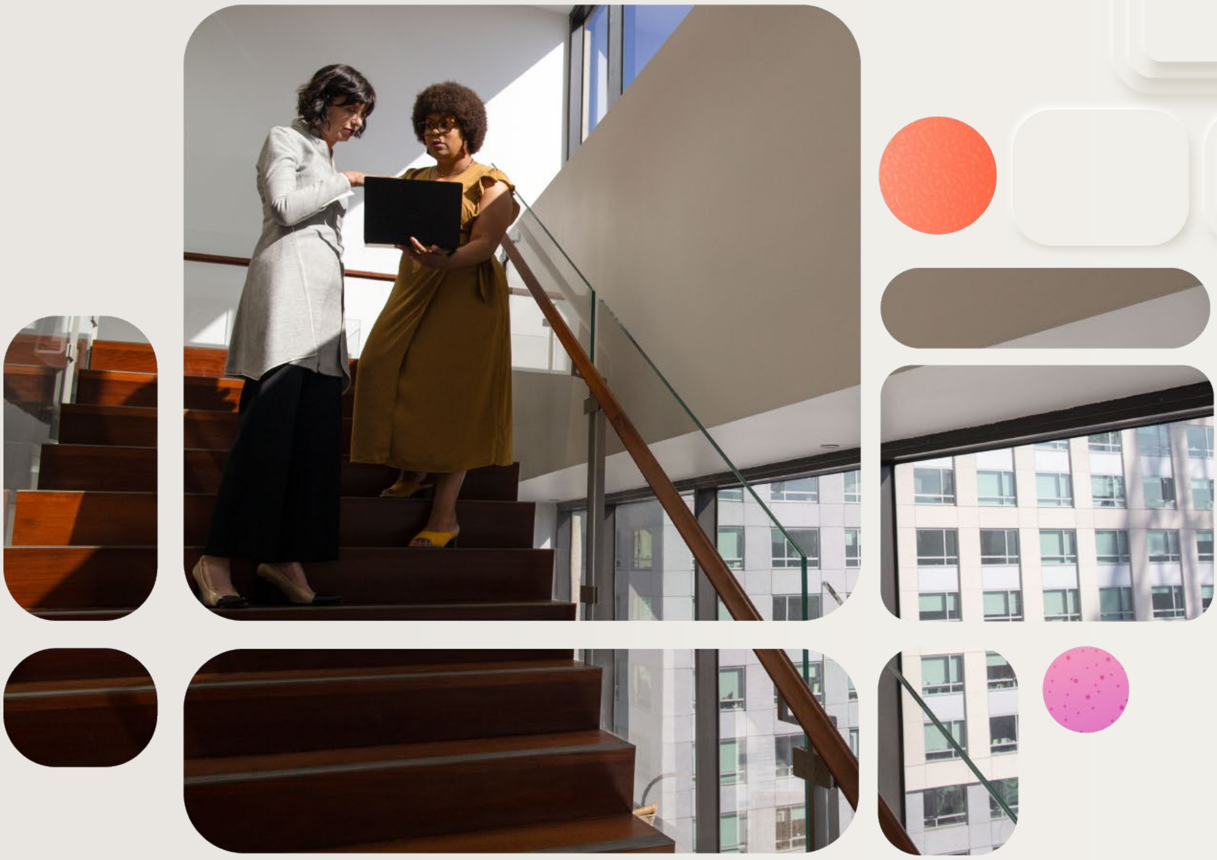
Microsoft

Microsoft Computer Science Curriculum Guide (MCSG)

Table of Contents

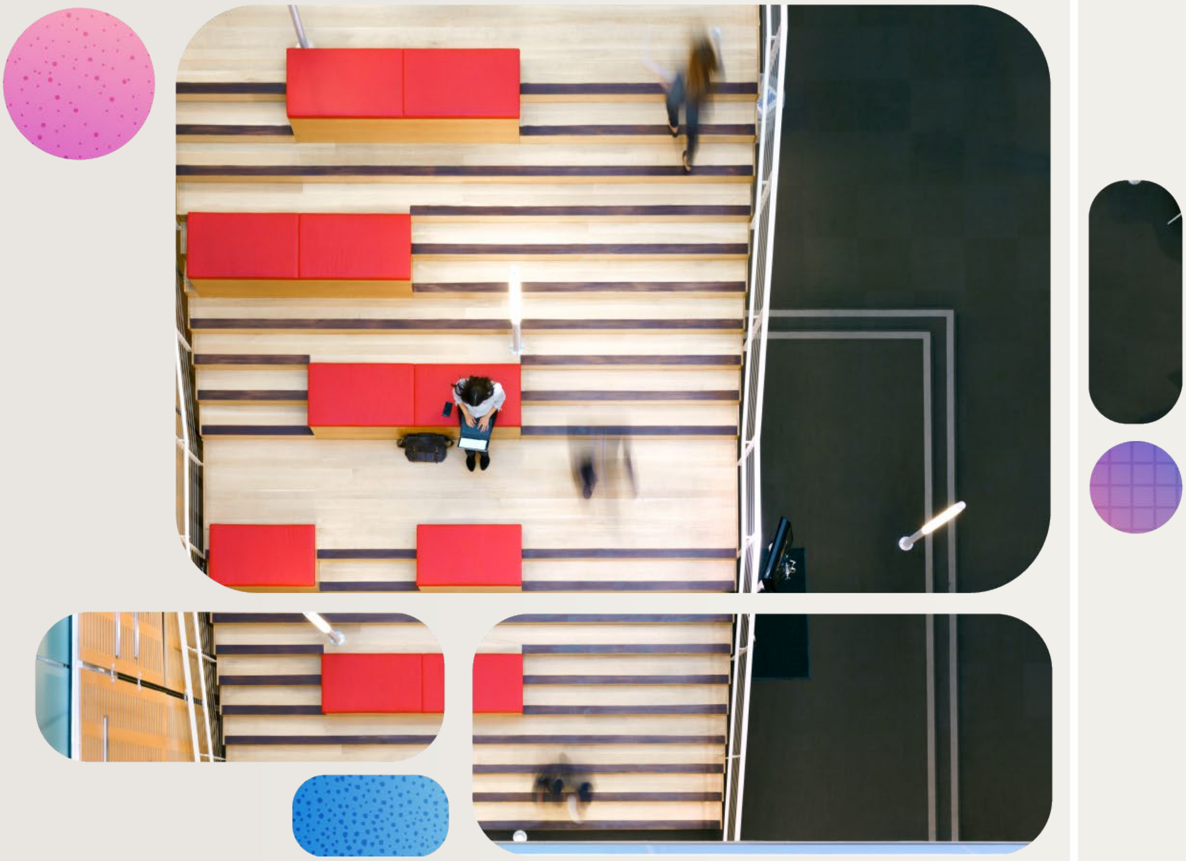


Objectives of this Guide

The end goal of this Microsoft Computer Science Guide (MCSG) is to ensure that students develop a range of competencies that enable them to pursue a computer science-related subject at tertiary education, enter the job market with appropriate skills, or become entrepreneurs. Microsoft believes that students should not just be consumers of technology but creators of it, and protagonists in the evolution of computing. By focusing on intriguing questions and diverse challenges based on real-world problems, the MCSG should encourage a diversity of students to participate in computer science courses. Graduating students should be able to build commercial computing solutions to address the issues they learn about.

To this end, the objectives of the MCSG are to:

- Provide information and recommendations for the development of computer science education standards for mainstream primary and secondary schooling. Every country will have different requirements, so the main purpose of the MCSG is to give ministries of education enough information and recommendations to frame and build their own version of the curriculum.
- Establish a future-proof framework for further development. By selecting Domains, Big Ideas, Big Questions, competencies, and topics that are at the cutting edge of CS, the MCSG will have longevity and allow ambitious skills development programs.
- Drive economic development. By 2024, roles requiring digital skills are predicted to grow by 12 percent[i]. In some countries, the goal of the MCSG will be to help address youth unemployment by encouraging young people become entrepreneurs, examples of which are often highlighted in the news.[ii] [iii]



Background to the MCSG

To develop the MCSG, Microsoft called on a group of internal and external experts with experience as computer science professionals, educators, and in academia. This group drew on Microsoft's expertise as a world-leader in technology and a decades-long employer of computer scientists to understand what content is important to include in a computer science curriculum framework. It consulted academic research in the learning sciences and curriculum development to design the structure and sequencing of the curriculum framework and to identify the right pedagogies to recommend for the content.

Further to this, the group conducted research into what other education systems around the world have been doing in terms of developing CS curricula at a school level. This research was conducted both as traditional desk research, in the form of a literature review, and as a discussion with ministry of education officials on computer science in the curriculum that was part of a live event hosted by Microsoft, European Schoolnet and STEM Alliance.[iv]

These research initiatives served to teach the MCSG designers both what has been successful and what has been learned about past large-scale CS curriculum designs and implementations. It also gave the writers of the MCSG a clear idea of Ministry of Education priorities in terms of content, competences, and objectives. All of this research, summarised in the sections that follow, enabled the design of the innovative curriculum framework that is outlined in this document.

Addresses Ministry of Education feedback

Feedback from Ministries of Education was collected from a live, online event co-hosted by Microsoft and European Schoolnet. During this event, representatives from the European Commission provided a summary of computing, computer science and computational thinking initiatives in various European countries. A panel and question and answer session was also held with representatives from the Ministries of Education in three European countries who had developed or were in the process of updating their national curricula to include computer science. Participants in the panel were asked about their countries' objectives in developing and implementing CS curricula and the outcomes they hoped to achieve. They discussed the barriers they had experienced, including those inherent in their own education systems, and support mechanisms they were putting in place at various stages in the process. Participants to the online event were able to ask questions to learn from the best practice and experiences of these countries.

One of the key takeaways from the event that was fundamental in the development of the MCSG was that computer science is not just about programming. One participant stated that her country's objective in developing a comprehensive CS curriculum was to create a "digital wisdom" amongst citizens. Government objectives in creating CS curricula were not necessarily about more students becoming computer scientists but giving more student access to the fundamentals of the subject so that they could use their skills in "known and unknown situations," as one participant said. Countries see the benefits of computer science as an outlet for students' creativity, problem-solving and innovation. They want to show young people what is possible in terms of their own future careers and in addressing problems facing our world today and in the future.

The MCSG also takes account of feedback from Ministries that creating an outcomes-based CS curriculum not only prevents it from becoming outdated or obsolete too quickly, but it provides more opportunities for students to be innovative. Other countries have found that overly prescriptive CS curricula that dictate the technologies to be used rather than the competencies to be achieved promote "teaching to the test" and do not allow students to be creative in their problem solving. Countries have had to provide support for teachers to enable them to feel confident with this level of flexibility in teaching complex computer science concepts.

Some of the barriers to adopting CS curricula that these countries experienced will also need to be addressed by any users of the MCSG. Participants expressed that traditional exams are often an inappropriate method to assess competencies in computer science. Indeed, the MCSG organises more advanced work into labs and projects that could be used to evaluate student learning rather than high-stakes exams that often measure student recall at a certain point in time. Learning projects and student portfolios, also mentioned during the panel discussion, allow a more in-depth examination of students' mastery of competencies and the progress they have made on their learning journey.

Finally, education systems face a challenge in finding enough teachers qualified to teach computer science and in keeping their professional development up to date with the constantly changing field. Support structures are needed to provide initial training and continuous development for a computer science teaching workforce.

Makes use of Innovative Pedagogies

The MCSG allows for flexibility in teachers' choice of pedagogies as well as technologies, but has as its core an inquiry-led, problem-solving approach. The MCSG poses "Big Questions" to students, which not only allow the curriculum to be flexible and adaptable to accommodate the rapid pace of technology change, but also encourage students to apply what they have learned to solve real-world problems.

There is abundant research on this kind of problem- or project-based learning (PBL) that shows a positive impact on students' development of skills and their motivation towards learning.[v] For example:

- Students who have participated in PBL retain more knowledge, acquire better-developed critical thinking skills, and have increased motivation to learn than those who learned through traditional methods.[vi, vii]
- PBL is positively associated with skill development in students, regardless of the stage of education or level of expertise of the student.[viii]
- PBL can benefit students considered "at risk" who may come from challenging home circumstances, for example. In mathematics education in particular, use of PBL has decreased the achievement gap between these at-risk pupils and their peers.[ix]

Most critically, PBL provides the context needed to develop collaboration, problem-solving, systems thinking, communication, and the broader competencies that are essential to a modern education. A PBL approach to curriculum simulates the work learners will experience in most kinds of work later in life. A PBL approach help them learn to work in teams and directing their own acquisition of new knowledge and skills.

One of the challenges of PBL is the quality of the initial question or problem that needs to be answered or solved, which can vary substantially depending on the teacher. A key feature of the MCSG is that it provides guiding and inspiring questions which students will learn competencies to address.

Fosters equity in computer science education

Across the world, there are challenges of equity in CS education. Issues of equity manifest themselves in terms of the availability of CS courses in all schools in a country and the level of participation in CS courses from diverse groups of students.

Research has shown that there has been relatively low participation in higher level CS courses among girls and some minorities. This can be due to availability of courses, as mentioned, but some research indicates that it is connected to the way that CS is taught and reinforced at school. For example, in the United States, a review of research uncovered several reasons why girls are often deterred from taking computer science, including:

- Teaching style that uses lecturing rather than hands-on, real-world projects.
- Lack of collaboration with peers.
- Lower levels of self-confidence, and lack of clarity around careers in computer science.
- A feeling that mistakes or risk-taking are not permitted in computer science.[x]

In those countries that have created new CS curricula, there is also a tremendous challenge in implementation: teachers feel insufficiently prepared to teach CS, and there are low levels of recruitment of CS graduates to the teaching profession. When CS professionals do become teachers, they often lack the pedagogical skills to understand how to teach the very topic they have used throughout their own careers.xi As a result, the need for in-service CS professional development is immense. Given these challenges, at global scale, there is a clear incentive for collective action to more effectively design CS curricula and prepare teachers appropriately to engage students in impactful CS learning.

The MCSG takes advantage of several approaches supported by the research literature to help make the teaching of CS more equitable to students of all backgrounds and abilities. For example, equity is overtly woven into the learning goals, concepts, and challenges in the MCSG. Students study the impacts of computer science on issues of accessibility as well as the positive and negative impacts of AI on society. The deliverables associated with many CS tasks provide opportunities for non-native language speakers to demonstrate competency without having to produce as much written work as they might in other subject areas. The project-based and inquiry-based learning approach uses real-world, socially relevant problems to help engage females and underrepresented minorities in computer science.

The MCSG takes into account that not all schools have adequate equipment to provide an "ideal" CS education. However, many topics, especially in primary schools, can be learned without computers. This curriculum makes many suggestions for "unplugged" learning: i.e. computing activities that don't require physical hardware. Block-based programming helps to make some concepts of programming and CS more accessible to younger children. The MCSG also provides a variety of approaches to each topic, accomplished by the various learning goals, concepts, and challenges offered, that will help meet the varied learning needs of students of all ages.

Finally, to reach the widest possible range of learners throughout their education, the MCSG offers three possible graduation pathways: academia, vocational or entrepreneurial. The MCSG believes that all learners should have access to a CS education, should they desire, regardless of their ultimate career trajectory.

Flexible framework encourages local adaptation

It is important to note that Microsoft is not providing a prescriptive, one-size-fits-all computing curriculum. Rather, the MCSG is exactly as the name suggests: a framework. A fully developed curriculum requires substantial creation and adaptation of schemes of work, classroom activities and a system of assessment, including the development of standards. This necessitates substantial input from local educators and education experts who understand the context of a country's education system and the conditions inside its schools.

However, what Microsoft is supplying with this framework is more than just a matrix of topics to be covered and their progression. The MCSG gives guidance on the topics that should be covered in an innovative and enduring CS curriculum and the sequence in which they should be covered. It does not provide high-stakes assessments but suggests projects and lab work that can allow students to apply their learning to demonstrate their mastery of the necessary CS outcomes. The MCSG does provide links to Microsoft and other high-quality training, programmes and learning resources that can be used by CS educators and students, but it leaves plenty of room for addition of local content and resources. In other words, the MCSG provides sufficient detail to be adopted by schools but is flexible enough for different countries to adapt to suit their context or preferences.

Gives students a foundation in CS while introducing them to cutting-edge content

Before developing this framework, research was conducted to identify global best practices to learn from what has already been done in developing CS curricula at a school level.

Where possible, the authors examined countries that provided curriculum spanning all ages across primary, secondary and further education, but in some cases, the computer science curricula were only offered in secondary school. In other countries, the computer science curriculum is part of a much larger Computing or Digital Technologies curriculum that also includes other technology-related topics.

Certain key topics were identified as being commonly used to organise the high-level content domains within the curricula. These include content domains such as:

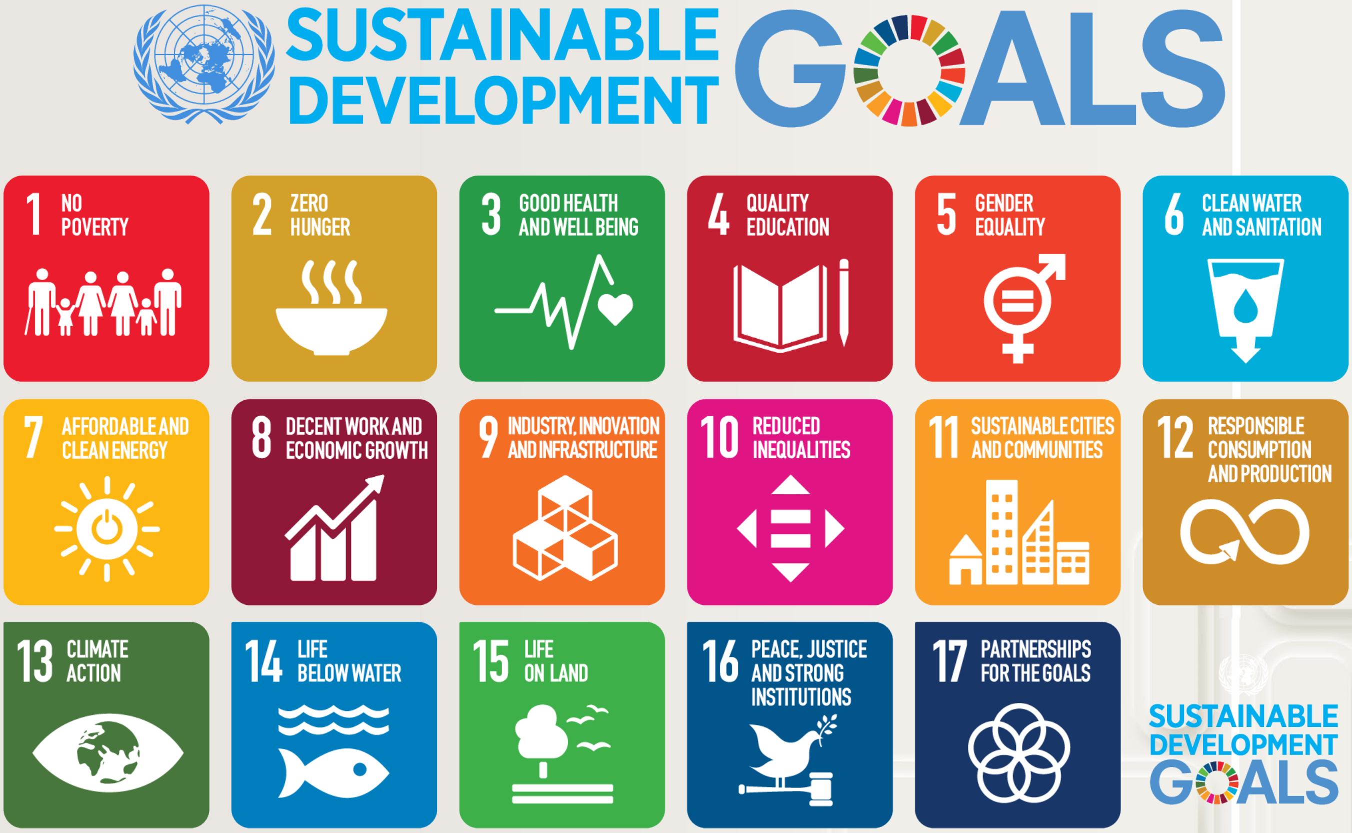
1. Computing systems: including system fundamentals, computer organisation and hardware
2. Data and data-related topics
3. Networks
4. Algorithms
5. Programming
6. Impacts: including cultural and social impacts of computing
7. Computational thinking
8. Problem solving: often a combination of computational thinking and algorithms.

The MCSG includes this core content so that even those countries who already have national curricula or standards in computer science might benefit from the Big Ideas, Big Questions or resources provided in the MCSG. However, the MCSG also includes content on such cutting-edge technologies, as artificial intelligence (AI), cloud, blockchain and Quantum, which, according to the research, are often omitted from other national curricula.

Provides real-world context by using the United Nations Sustainable Development Goals

As part of the United Nations 2030 Agenda for Sustainable Development, member countries agreed on 17 Sustainable Development Goals (SDGs) which require countries to work together to solve the most urgent problems facing our world today. The 17 SDGs are related to each other – none can be achieved in isolation – and are present in some form in developed and developing nations alike.

The MCSG takes advantage of the UN SDGs as themes for the Big Questions across all Domains in Phases 3 and 4. Solving the SDGs will require innovative, global solutions, and the MCSG aims to prepare students to tackle challenging problems such as these with cutting-edge technology and superior problem-solving and analytical skills. Furthermore, by connecting CS learning objectives to real-world problems, the MCSG hopes to engage learners in the topics and inspire them to develop solutions to real-world problems rather than simply completing abstract tasks.[xii]


SUSTAINABLE DEVELOPMENT GOALS
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

Specifically, SDGs have been mapped against each of the Big Question, Sprint and Pitches in the project and business phases. Students are expected to research the given SDG and explore problems that can be solved with CS. They need to demonstrate an understanding of how CS can make a positive impact towards the SDG if their proposed solution was to be implemented and scaled in the real world. To this end they will need to identify and specify target segmentations and use cases. Ideally, students will build prototypes that demonstrate creative and realistic solutions.

Takes advantage of Microsoft expertise, content, programmes and support resources

Through Microsoft’s work in primary, secondary and tertiary education and with computer scientists and programmers, a vast array of content, training and other support resources are available for teachers and students of computer science at all levels – these are listed in the section entitled “Content Used in the MCSG” towards the end of this document.

The MCSG provides a broader vision of CS – one that allows students to acquire an inspiring 360-degree experience of the world of computing, and enables them to acquire the broader set of skills required to innovate with technology as an outlet for their creativity. The wealth of accompanying resources from Microsoft help make this possible.

Implementing the MCSG

> Implementing the MCSG is to a large extend about making a series of decisions. This implementation guide will step you through the decisions that you'll need to make and offer suggestions for different options.

Implementation options

The first decision that needs to be made is whether the implementation planning will be done internally by your organisation or supported by Microsoft.

Microsoft Education Industry Services

If you need support from Microsoft, you may decide to engage Microsoft Education Industry Services to deliver some or all the following:

- Change Management
- Program design and implementation through Microsoft's Education Transformation Service (ETS)
- Gain actionable insights around delivered content using Microsoft's Open Education Analytics (OEA) solution
- Content creation, curation, delivery, and analytics solution using Microsoft's Online Curriculum Hub (OCH) solution.

Education Global Training Partners

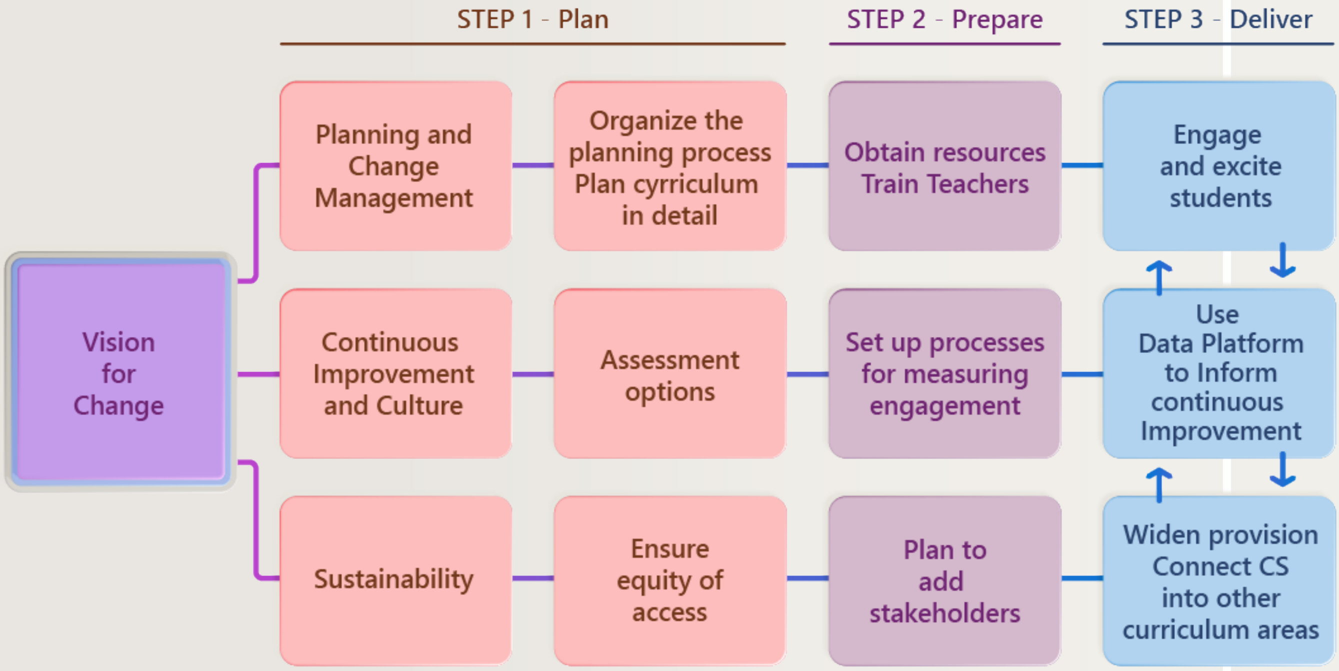
You can also make use of Microsoft's Global Education Training Partner Program whose members have undergone an official Microsoft training program enabling them to deliver in-depth learning experiences aligned to the Education Transformation Framework on behalf of Microsoft. The program offers professional development experiences in one or several of the 5 following areas:

- Education Transformation Framework
- Modern Classroom Teacher Academies
- Future-Ready Skills and Industry Certifications
- Minecraft Education
- Hacking STEM

Step 1 – Plan

Organise the Planning Process

Implementation planning has three steps – plan, prepare and deliver: and three strands – planning & change management, continuous improvement and culture, and sustainability.



	STEP 1 - Plan		STEP 2 - Prepare	STEP 3 - Deliver
Vision for Change	Planning and Change Management	Organize the planning process Plan cyrriculum in detail	Obtain resources Train Teachers	Engage and excite students
	Continuous Improvement and Culture	Assessment options	Set up processes for measuring engagement	Use Data Platform to Inform continuous Improvement
	Sustainability	Ensure equity of access	Plan to add stakeholders	Widen provision Connect CS into other curriculum areas

Let us now unpack each of these one step at a time.

Planning needs to be set up as a process that covers the full spectrum of what it takes to implement CS in schools, and includes the following:

Vision for Change – Define a vision that specifies the intended future state of the organisation.

Establish and agree the ideal future state. Make it bold, ambitious, motivating, and achievable

The word "curriculum" is derived from the Latin word for 'racecourse' – the course of learning experiences through which children grow to become mature adults. As with a physical racecourse, its quality and design are going to have a huge impact on the effectiveness of those racing on it. The curriculum defines the learning outcomes required from an aspect of schooling, so setting a clear and motivating vision is an imperative first step.

Visions for Implementing Computer Science can focus on, for example, economic development, addressing inequality, digital empowerment, or a combination of these.

“A world-class education that enables all citizens to thrive in an ever-changing technological world”.

Often a vision can focus on modernising from the old-style ICT curriculum, about using computers, to Computer Science which is about developing computing solutions.

Whatever the vision, it needs to reflect your organisation’s values, sustainability, context, pedagogical approaches and needs and all stakeholders should have a say in it.

Goals – Specific, time-bound and measurable outcomes.

Establishing what stakeholders need from the CS curriculum:

- Who are the stakeholders?
- What do they need?
- What is CS for?
- What should students learn?
- Economic outcomes
- Social outcomes

Goals needs to align to the vision, they need to be concrete, clear and time-bound. For example: “Reduce the number of unfilled vacancies that require Computer Science by 50% in 5 years”.

Policy – Clearly define the stakeholders and required outcomes for each stakeholder group. Guide the decisions that will achieve the required outcomes through:

- Legislation – legally binding obligations, e.g. is the curriculum compulsory?
- Intentions – high level goals and statements of values
- Procedural – state what needs to be done and by whom
- Guideline: Non-binding information that assists in implementing procedures

Priorities – focus resources and investments e.g.:

- Prioritised curriculum focus areas
- Decide on how to phase-in CS
- Targeting the neediest
- How to optimise the use of time
- Using data and analytics to continuously improve
- Hire and develop the best quality teachers
- Accelerate and personalise learning
- Promote CS learning beyond the classroom
- parents and the community

Strategy – Plans of action designed to achieve specific goals

Budgets – Allocations of capital and resources to operations and projects

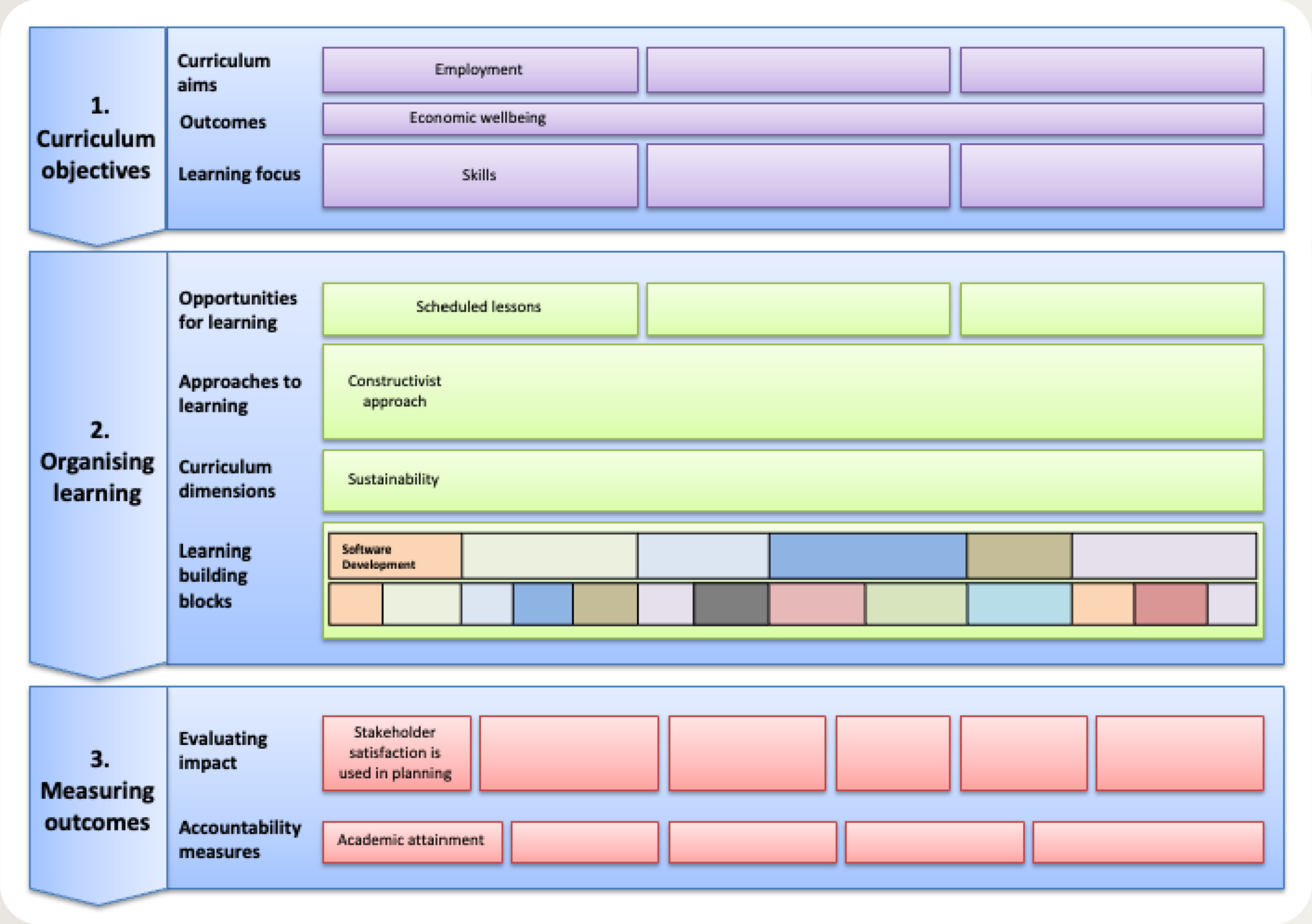
Architectures – Organisational, information and technological systems, and physical environments

Actions – Prioritised organisational objectives; workgroup and individual performance goals; culture and group processes

Plan the Curriculum in Detail

There are several Computer Science standards that are available ‘off the shelf’, such as CSTA and CAS, and these can be used verbatim or adapted to your needs. The Microsoft Computer Science Framework is mapped to both CSTA and CAS standards but go beyond – for example by pointing to supporting content – to provide a flexible, up-to-date framework within which you can build your curriculum.

Whatever route you choose, it helps to summarise your curriculum plan in a single frame. The example below includes some sample features.



A key decision to be made is about the phasing-in of the implementation. For example, a time-based phased implementation by grade or stage, or geographically phased implementation, or a needs-based implementation such as prioritising schools with specific challenges. Whichever implementation model is chosen, a key point to bear in mind that early, well publicised, small-scale successes can help win advocates for a bigger implementation.

Another key decision to be made is whether to have a standardised curriculum across a schooling system or not. On the one hand, there are strong arguments for having a curriculum that is open-ended and adapted to local needs, as opposed to it being linear and closed. On the other hand, there is equally a strong argument for having a curriculum that is sufficiently well defined for national or state level consensus about what counts as mastery in the subject area. It is, however, possible to have both. A core curriculum based on standards that everybody agrees on can be applied at country level, whilst at the same time, schools can be given the freedom to add more open-ended and localised extensions. This is where a framework approach, such as MCSG, has benefit.

Whichever route is taken, implementing a curriculum at national level requires an organisational taxonomy including:

Time – stages of development

What students will learn – time-bound programmes of study and specified learning activities

Proof of learning – attainment targets linked to stages

Levels of learning – concrete descriptions of different levels of mastery

Sequence – every learning activity needs to be placed and sequenced within a framework that encompasses the entire CS curriculum.

Learning & Teaching narrative – how will the curriculum be delivered

Competencies

For Computer Science to be valuable to both individuals and society, the curriculum needs to be based on competencies rather than just memorised facts. The core process of Computer Science is developing solutions, so Computer Science is about the acquisition and application of skills, and this requires a competency orientated curriculum where the learning process resembles this approach –

> Inspiring problem to solve → aspiration → goals → do learning task and demonstrate new competencies → get recognition for competencies

Assessment Options

In the Internet era, when information is so readily available, mass scale memory tests as a primary means of assessment is irrelevant and represents poor value for money. However, to be competent in Computer Science, students will need to have learned some basic facts and CS vocabulary. This means that assessment should cover both competence and knowledge.

Formative, or Low-Stakes Assessment

Low-stakes, or formative assessment, can have powerful effect on the development of skills and knowledge[xiii]. It can help teachers to identify learning gaps which need addressing and can help children to identify their own learning gaps and strengths. Formative assessment can give students regular feedback and insight into their progress to help develop independent study and self-assessment skills.

Activities that can be part of formative assessment include:

- Completion of individual problem-solving challenges
- Contribution to group problem solving challenges
- Multiple choice quizzes or tests
- Asking children to re-call key CS concepts and terms
- In-class reviews
- Observed practical work
- Exhibition and presentations

Certification

Students who gain professional level certification gain confidence and a sense of personal accomplishment[xiv]. Certification also gives students credibility to enter the IT jobs market, where there are significant demands for certificated skills. In the United States, certification can also be added to transcripts for institutions to enable learners to obtain college credit or advanced placement in a degree program[xv].

Details on the kinds of certification available, and how they fit into the overall Computer Science curriculum are covered in the "Microsoft Certification" and "Certification Pathways" sections below.

Summative Assessment, or High Stakes Assessment

Most schooling systems funnel students towards a terminal examination which acts as a ‘filter’ for higher education. Given that Computer Science can be pursued both academically and vocationally after formal schooling, assessment needs to culminate in a standard that is acceptable to both employers and universities.

Employers need people with skills[xvi], whilst universities tend to want to take on students who are equipped with the *knowledge* necessary to progress successfully[xvii]. But the assumption that students will automatically select either a vocational or academic path needs to be challenged because increasing numbers of students are able to start on the road to entrepreneurialism through mini businesses set up in school[xviii]. This requires a third dimension to assessment that gives at least some students the tools to develop mini technology start-ups based on their solutions to practical problems, and their response to a business application of CS challenge.

So how can we reconcile these different requirements?

Given these requirements, a hybrid approach of written paper plus portfolio plus written exam could be considered. Instead of testing everything together in a big exam at the end of the course, students should be given the opportunity to show what they’ve learned throughout their CS studies, culminating in a computing solution to a real-world problem framed in a commercial context. This means they can build on their achievements and get opportunities to learn and grow as they progress, without the need to cram for a big final exam.

By the end of following a CS curriculum, students should be able to:

- Demonstrate that they understand key concepts and principles of Computer Science
- Demonstrate key Computer Science skills
- Analyse real-world problems in computational terms
- Apply their understanding to make reasoned judgements and build optimal solutions
- Frame a computational solution in a commercial context

A summative assessment could comprise of the following:

- A course portfolio comprising lab logbooks, code samples, digital prototypes, research reports; assessed by teachers and validated by a recognised standards authority.
- A written examination can be used to ensure that key CS concepts are understood.
- A capstone project – a computational solution framed in a commercial context and assessed by presentation and documentation, and like the course portfolio assessed by teachers and validated by a recognised standards authority.

Combined, and weighted according to stakeholder requirements, these assessment elements can provide a summative grade for each student.

Ensure Equity of Access

Equity is about recognising that each person has different circumstances and allocates the exact resources and opportunities needed to reach an equal outcome. Top performing schooling systems tend to have strong equity. In OECD countries, most equity driven reforms are targeted at “at-risk” groups, whereas in the developing world they tend to be targeted at low-income groups, women and rural populations.

But equity cannot be achieved by treating everyone equally. Clearly, every student, teacher and school are different – so treating them all in exactly the same way isn’t going to promote equity. To achieve equity, we need to treat each school and each student, differently as needs require. Good use of data gives us the means to accurately adjust levels of different resources as each school and each student requires[xix].

Some general principles for driving equity include:

1. Make sure all students have equal access to the CS education.

Whilst this seems obvious, there is much to do to ensure equal access including time for CS on the timetable; skilled and motivated teachers; classrooms; computers; networks and internet access; content; assessment and accreditation – at the least, all these should be available equally to all students.

2. Give students access to computer science education early.

Engaging children early on is another way to drive equity, and this can be achieved in a number of ways. For example, Block-based programming languages such as MakeCode allow students as young as 5 years old to program without the obstacle of wrongly typing commands. Also, some introductory aspects of Computer Science can be learned through gamified kinaesthetic experiences without a computer – 'unplugged'.

3. Make special provision for those with special needs

Impairments don't necessarily stop people becoming good Computer Science practitioners if they are given the right tools and support. For example, keyboards and mice are no longer the only way to input and manipulate code, text, or produce diagrams, and screen readers can support students with visual impairments.

4. Reaching Females and Underrepresented Minorities

Introducing students to diverse role models from the world of computing, orienting the curriculum towards solving relevant real-world problems, focussing on technologies which they regularly use, and engaging family and community are all ways in which females and underrepresented minorities can be engaged.

5. Allow students to express individuality through coding.

Students will feel more interested in Computer Science when they realise that it can be a vehicle for them to express their individuality[xx]. To engage all students, it's crucial that they understand that Computer Science is relevant to them regardless of their future career choices and allows them to showcase their unique interests.

> *"If you're a great graphic designer, if you're a great web designer, if you want to create great apps or NFTs – all of that is done through tech and understanding of tech"*[xxi].

Step 2 – Prepare

Obtain Resources

Purposeful engagement is the key to choosing resources to deliver your curriculum. Every learning task should be different, and therefore requires learning content with different features – form follows function.

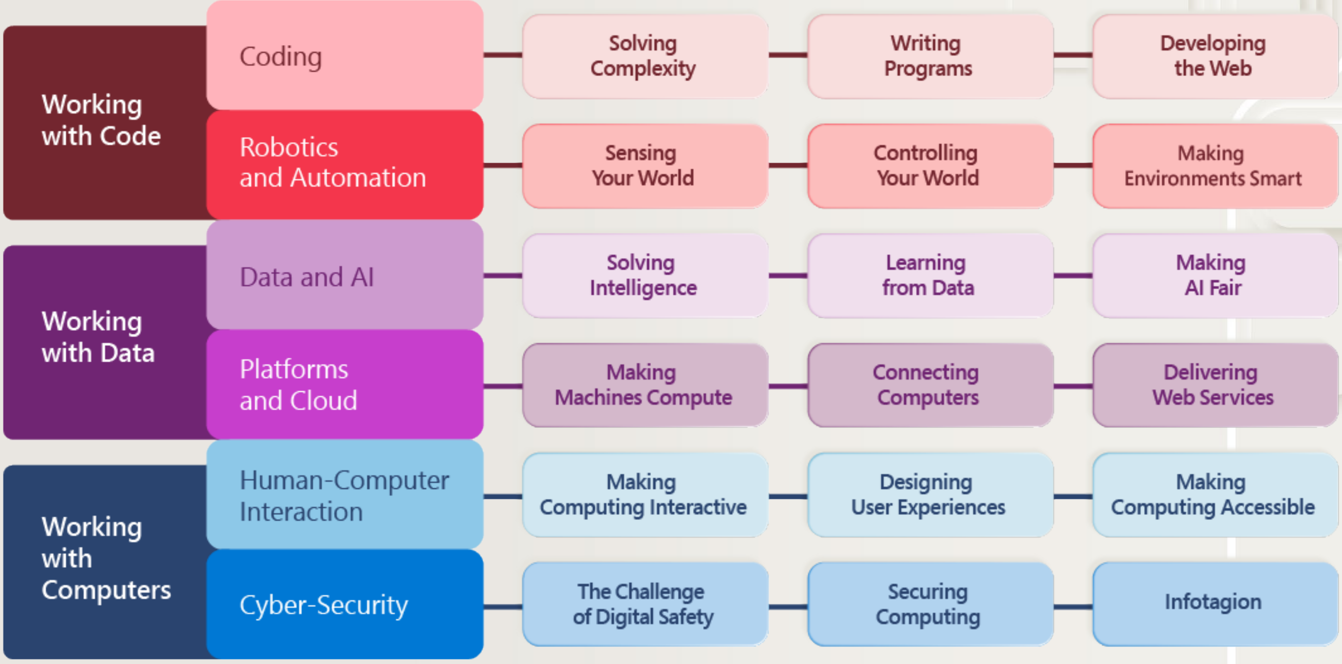
Learning resources should be engaging, hands-on, and have interactive content that allows students to participate fully, work creatively, and think critically throughout the learning process.

Engagement increases with content that is current and relevant to students' lives. High-quality video and images matter. Additionally, engagement is increased on digital platforms designed to facilitate both independent and collaborative learning and when students are given opportunities to create, rather than simply consume information[xxii].

"Off-the-Shelf" Solutions

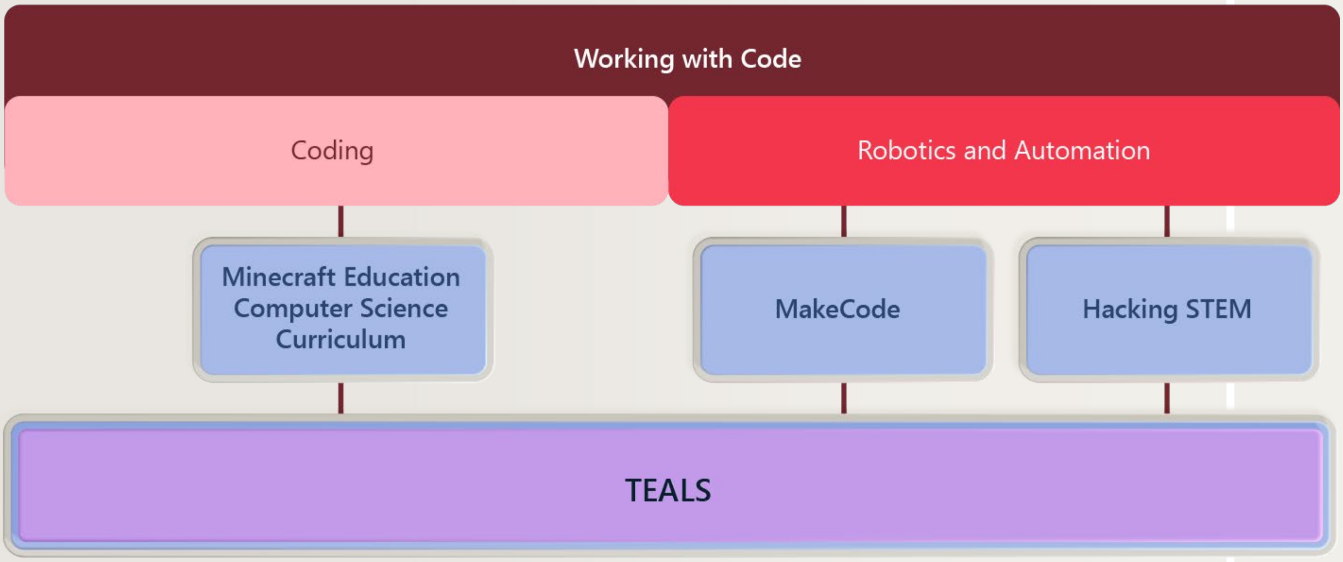
For those organisations wishing to make a quick start with either upgrading to modern Computer Science or implementing CS for the first time, there are three Microsoft programs that can both help immediately and become part of a broader and longer-term framework based on the Microsoft Computer Science Guide.

The Microsoft Computer Science Guide has 6 subject domains covering all phases of formal education. Within each domain are 3 'big ideas', which provide topic-based structure for sequential learning.



Working with Code	Coding	Solving Complexity	Writing Programs	Developing the Web
	Robotics and Automation	Sensing Your World	Controlling Your World	Making Environments Smart
Working with Data	Data and AI	Solving Intelligence	Learning from Data	Making AI Fair
	Platforms and Cloud	Making Machines Compute	Connecting Computers	Delivering Web Services
Working with Computers	Human-Computer Interaction	Making Computing Interactive	Designing User Experiences	Making Computing Accessible
	Cyber-Security	The Challenge of Digital Safety	Securing Computing	Infotagion

For those organisations wishing to get quick results, the first two domains in the MCSG – Coding and Robotics – could be prioritised.



Working with Code

Coding

Robotics and Automation

Minecraft Education Computer Science Curriculum

MakeCode

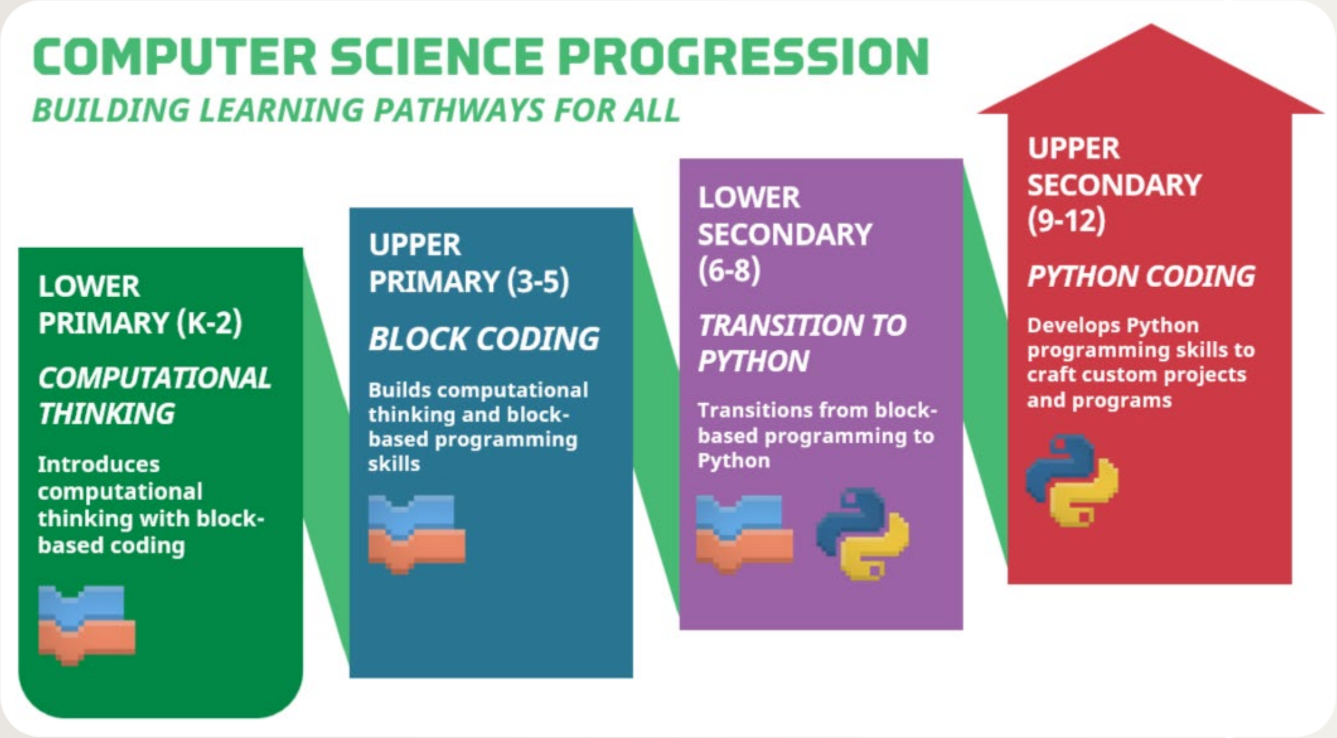
Hacking STEM

TEALS

Minecraft Education Computer Science

Minecraft Education offers 150 hours of Computer Science content for beginner to advanced levels supported by a range of training material for educators. The material has been exclusively designed to give teachers a consistent and easy-to implement curriculum resource.

Students can start with Minecraft coding activities from the Hour of Code, coding with blocks or Python to solve puzzles in engaging Minecraft worlds. They can then move onto standards-based Lesson Collections that support learners at every stage of their Computer Science journey.



COMPUTER SCIENCE PROGRESSION

BUILDING LEARNING PATHWAYS FOR ALL

LOWER PRIMARY (K-2)

COMPUTATIONAL THINKING

Introduces computational thinking with block-based coding

UPPER PRIMARY (3-5)

BLOCK CODING

Builds computational thinking and block-based programming skills

LOWER SECONDARY (6-8)

TRANSITION TO PYTHON

Transitions from block-based programming to Python

UPPER SECONDARY (9-12)

PYTHON CODING

Develops Python programming skills to craft custom projects and programs

Teachers do not need to have any prior computer science experience and professional development support for teachers can be found in the Minecraft Education Teacher Academy and Webinar Series.

Follow these links to explore Minecraft Computer Science Resources:

- Resource guide – https://education.minecraft.net/en-us/resources/computer-science
- Curriculum Guide – https://education.minecraft.net/content/dam/education-edition/software-downloads/Minecraft_CS_Curriculum_Guide_0622.pdf
- Teacher training – https://education.minecraft.net/en-us/resources/get-trained

MakeCode

Microsoft MakeCode is a free, open-source platform for creating engaging computer science learning experience and provide a progression pathway into real-world programming.

The name “MakeCode” comes from Make & Code – i.e. linking coding to physical objects and systems. MakeCode therefore is ideally suited to the MCSG Robotics domain, where the goal is to learn how to use code to manipulate the physical world, including the Computer Science subject of Internet of Things.

MakeCode is ideal for project based learning as it works seamlessly with a range of hardware including the BBC micro:bit and Adafruit microcontrollers allowing student to build a wide range of products and levels of complexity. For example, beginners could use MakeCode’s block-coding capabilities to create musical instruments, whilst more advanced level students could use MakeCode’s JavaScript capabilities to control mini vertical farms.

Several types of resources are available to support teachers with the use of MakeCode –

Intro to CS with the micro:bit – a free, 14-week introductory course for students ages 9-15 that covers basic computer science concepts like binary numbers, variables, iteration, and conditionals. The course focuses on making as well as coding and lessons include projects such as a micro:pet, a board game.

Intro to CS with Minecraft – an entry-level computer science course based on MakeCode for Minecraft with lessons that contain 'unplugged' exercises, guided coding activities, and a self-directed project that maps to CSTA standards.

Maker Course for the Adafruit Circuit Playground Express – a project-based course includes 12 different fun and easy making and coding projects using the Circuit Playground Express microcontroller. The projects are scaffolded and aligned to academic subject areas, with videos and step-by-step instructions.

Follow these links to explore MakeCode Computer Science Resources:

- Introduction to MakeCode – https://www.microsoft.com/en-us/makecode/about
- Curriculum and projects resources – https://www.microsoft.com/en-us/makecode/resources

TEALS – Technology Education and Learning Support

Microsoft TEALS is a Microsoft Philanthropies program that partners directly with individual schools to bring teachers and volunteers together to directly help students work with code. The goal of the program is to build teacher capacity and student interest in computer science with the goal of enabling schools to sustain equitable CS programs on their own. TEALS offers curriculum and support for students of all demographics and backgrounds and focuses on a culture of volunteerism in the tech industry so that teachers get the support they need.

Follow this link to explore the TEALS program – https://www.microsoft.com/en-us/teals/about

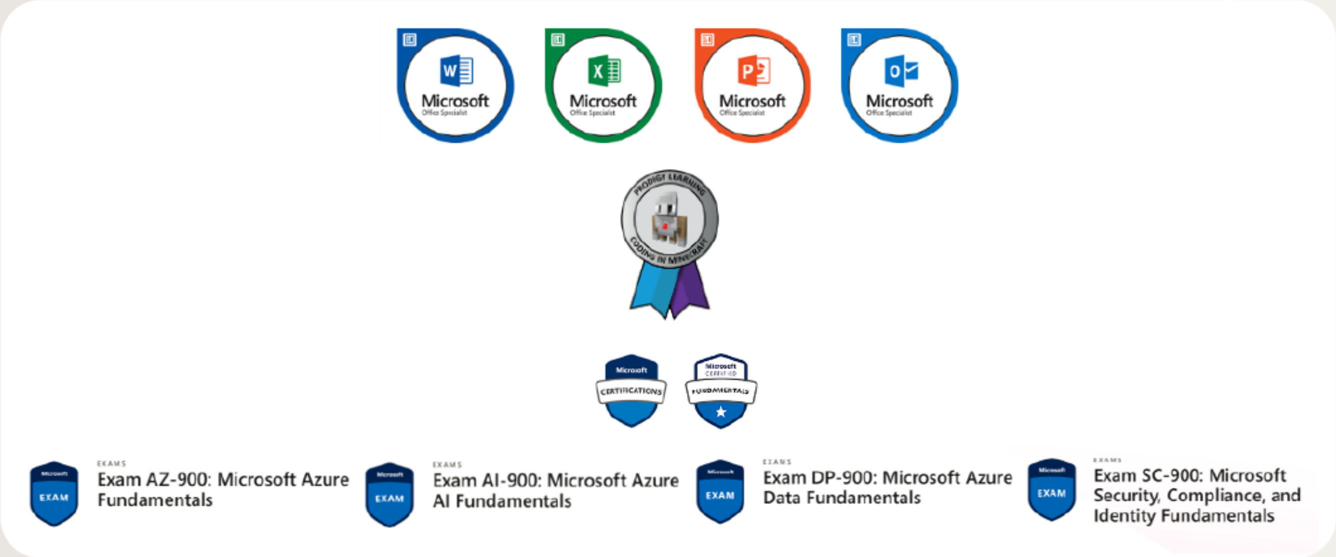
Microsoft Hacking STEM

Written by teachers for teachers Microsoft Hacking STEM provides middle school standards-based, project-based lesson plans focusing on visualising data across science, technology, engineering, and math (STEM) curriculum.

Microsoft Certification

Demand for technical industry recognized certification is growing across the world[i], and schools are a great place for learners to start their professional certification journey. Many school-aged students have gained Microsoft professional certifications and Microsoft offers the following certifications at primary and secondary school levels:

1. Microsoft Office Specialist
2. Coding in Minecraft
3. Azure Fundamentals


Microsoft Office Specialist
Microsoft Office Specialist
Microsoft Office Specialist
Microsoft Office Specialist
PRODIGY LEARNING
CODING IN MINECRAFT
Microsoft CERTIFICATIONS
Microsoft CERTIFIED FUNDAMENTALS
Microsoft EXAM
EXAMS
Exam AZ-900: Microsoft Azure Fundamentals
Microsoft EXAM
EXAMS
Exam AI-900: Microsoft Azure AI Fundamentals
Microsoft EXAM
EXAMS
Exam DP-900: Microsoft Azure Data Fundamentals
Microsoft EXAM
EXAMS
Exam SC-900: Microsoft Security, Compliance, and Identity Fundamentals

Train Teachers

There are not enough teachers able to teach CS, and a lot of anxiety amongst those teachers who could potentially teach the subject. A common fallacy is that students know more than the teachers. Teacher training, both pre-service and continuous professional development, is therefore critical to the successful implementation of CS. Whilst a lot of focus is put on subject matter expertise, it's worth bearing in mind that teaching CS is predominately about teaching and the development of thinking skills, and for teachers to be successful at teaching CS they need the abilities discussed below:

Problem-solving: A good CS teacher will be skilled at presenting and framing problems for students and guide them towards the solution through questioning. Teachers need to be able to teach students how to identify and categorise problems, conduct focussed research, and use a range of creative processes such as brainstorming to solve the problem.

Computational thinking: CS teachers need to develop their students' computational thinking skills which is about taking a complex problem and breaking it down into a series of small, more manageable problems. Each of these smaller problems can then be looked at individually[xxiii].

Algorithmic thinking: Algorithmic thinking solves the problem of how to build a replicable process – an algorithm – which is a formula for calculating answers, processing data, or automating tasks. Algorithms are basis of computer code but are encountered across a range of activities outside Computer Science, such as making a recipe for a meal, choregraphing a dance, doing long division, or writing music.

Design thinking: Design thinking starts with asking: 'why is this a problem?' In CS, the process iterates and narrows down possible solutions and ends with a working computing solution. Design thinking is conceptual, and its outputs are unique.

Critical thinking: Critical thinking is the ability to solve problems and make decisions based on logic and reasoning. As a computer science teacher, your ability to think critically can help you develop engaging lessons, solve technical problems, and encourage your students to think critically as well.

Other key abilities that CS teachers need to have include:

Communication: As with any subject, teachers should be able to explain complex topics in an easy-to-understand way and use various communication methods, such as whiteboards, software, and video, to convey their ideas.

Collaboration: In the world of work, CS is a team activity – sometimes teams of thousands of people will work on a CS project. So, it's essential that teachers can organise the learning process so that students get the opportunity to work in teams as well as individually.

Technological skills: CS teachers need to have mastery of CS subject matter, the ability to know what counts as good CS learning outcomes, and how to manage projects with a technical dimension. Fortunately, there is a lot of great material available that teachers can use.

Microsoft Teacher Training Support

Microsoft offers a wide range of support for teachers.

Online and live training is available across a range of subject areas and learning pathways including, for example:

- Introduction to coding
- Computational thinking and its importance in education
- Teach AP Computer Science Principles with the Microsoft MakeCode curriculum

Programs offered by Microsoft include:

Programs for Teachers – Microsoft Learn for Educators Program – online learning pathways that teachers can bring into their lessons.

Microsoft Educator – A global network of educators trained on using Microsoft tools in the classroom. By successfully completing the required learning path, participants are awarded their ME badge.

Global Minecraft Mentor – Minecraft Certified Teachers who have been accepted to a global community of educators passionate about Minecraft in the classroom and dedicated to support others in their Minecraft journey.

Follow these links to explore the Microsoft support for teachers -

- Teacher support overview – https://learn.microsoft.com/en-us/training/educator-center/
- Training and community overview – https://www.microsoft.com/en-gb/education/educators/training-and-community
- Calendar of training events – https://www.microsoft.com/en-gb/education/training-and-events/events-calendar
- Microsoft Learn for Educators program – https://learn.microsoft.com/en-us/training/educator-center/programs/msle/

Set up Processes for Measuring Engagement

Student engagement is predictive of key student success metrics including test scores, dropout rates, and graduation rates. Ongoing, data-based measurement of student engagement levels can reveal actionable insights to help schools. In 1993, Skinner and Belmont defined student engagement as "sustained behavioural involvement in learning activities accompanied by positive emotional tone.[xxiv]" In the always-connected world of social media and smartphones, student's attention needs to be greatly valued and focussed onto engagement in learning tasks. So how can we measure engagement?

The most obvious way is to ask students to report on how engaged they feel on a given learning task – what they like and what they would improve. Other methods include quizzes and polls, and participation surveys.

There IT tools that can be used to monitor and provide insights into student engagement and digital activity. Additionally, there are IT tools for monitoring social and emotional learning and checking-in on well-being. IT tools can be used to trigger notifications to educators and parents when students disengage from learning or when successes in new learning contexts are achieved and can be celebrated.

Plan to Add Stakeholders

The more supporters you have for Computer Science in the curriculum, the more sustainable it becomes, and the better the chance of increasing the subject's 'footprint'. So part of the preparation stage needs to be focussed on expanding the circle of stakeholders involved in CS in the curriculum.

School stakeholders are not only the school board, parents, staff, and students, but also local business owners; community groups and leaders; professional organizations; potential enrolments; youth organizations; the faith community; media, etc. Anyone who affects or is affected by what students learn can be considered a stakeholder, and these can be categorised as follows:

Core

- Students
- Parents
- Teachers
- Principals
- Local Community

Funding Providers

- Local Education Authority/State
- National Government
- Voters/Taxpayers

Key Influencers

- Non-Government Organisations (NGOs)
- Employers
- Investors
- Higher/Further Education
- The Media

Step 3 – Deliver

Engage and Excite Students

Even if CS is made compulsory, it's crucial to get students' excited and engaged about CS, and there a wide range of activities that can help –

Mini exhibitions and demos – take over a prominent space in the school during lunchtimes or breaks and exhibit real-world CS solutions using robots, Micro:bit, Raspberry Pi, and Internet of Things.

Hour of Code – annual, worldwide event that demystifies CS by exposing students to the basic steps in writing a computer program.

After school clubs – give enthusiastic students and hobbyists an opportunity to develop and show other students their CS creations.

Guest speakers – students usually respond well to a guest that is actually working in CS. The vast range of work that now involves computing means that there should be plenty of parents and school connections to draw speakers from. Having a regular guest speaker slot covering a range of roles increases the chances of students finding inspiration.

Competitions and prizes – students having their work praised in public is highly motivational to not only the student but to their peers as well, even more so if the prize is given by a respected external organisation.

Field trips – escaping from the confines of the classroom and learning about real-world applications of CS and they kind of work that CS practitioners do can really spark imaginations and inspire students to pursue careers in computing.

Use a Data Platform to Inform Continuous Improvement

Once you have an established curriculum up and running, you next need regularly review its impact on teaching and learning, making any adaptations or changes you need to improve it further. Leaders need visibility across their systems into who and where engagement and learning is happening effectively, and where it is not. Monitoring systems can provide reports that correlate well-being measures with digital and physical attendance or engagement.

Widen Provision

Assuming that implementation has been phased-in, early successes need to be publicised, lessons need to be learned and learnings need to be incorporated into wider provision. CS is important enough to have in every school at every level with equity across entire countries. Widening the provision is not limited to schools though – there is room across further and higher education, and the workplace for more CS education, so ambitions should not be limited to just a single sector.

Connect CS into Other Curriculum Areas

Computer Science integrates well with subjects across the curriculum. It demands both logical discipline and imaginative creativity in the selection and design of algorithms and the writing of programs; it relies on an understanding of the rules of language at a fundamental level; it encourages an awareness of the management and organisation of computer systems; it extends learners' horizons beyond the school or college environment in the appreciation of the effects of computer science on society and individuals. As well as integrating well as a subject alongside other subjects, there is plenty of room within other subjects for CS.

In the same way as computing is now infused into most kinds of human activity, so computing should be part of most aspects of the curriculum in one way or another. In the sciences, for example, standard practice in experimentation involves connecting sensors to computers and analysing the data computationally. The part of math that is taught in school – calculation by hand – isn't just tedious, it's mostly irrelevant to real mathematics and the real world, and a more appropriate approach for the times we live in is to teach math through computer programming[xxv]. Geography is a subject that is rich in data that can be analysed through computation. Algorithms are used across a wide range of subjects, for example even learning a new sports skill or rules of a game in physical education has an element of algorithmic thinking. The cross-curricular opportunities for implementing Computer Science into other subjects is endless.

So, a logical next step here in the implementation process is to audit the wider school curriculum to find opportunities of integrating elements of Computer Science to improve outcomes both in other subjects and Computer Science itself.

Organisation of the MCSG Curriculum

The structure of the MSCG was developed to adhere to the tenets of what the academic literature says about quality instructional design and curriculum development, as well as what learning sciences research tells us about how students learn.

Guiding the organisation of the MSCG is the principle from learning sciences research that all learning should build on learners' prior knowledge[xxvi]. Thus topics are arranged to provide a foundation for learners in early years and build upon that knowledge as the student's journey through the curriculum progresses. The real-world applicability of the questions the MCSG poses allows learners to relate to what is being taught by being able to compare topics to things they already know and understand from their own life experience[xxvii].

The MSCG follows the "Understanding by Design" (UBD) curriculum design framework[xxviii] that is thought of by many as one of the standards for high-quality curriculum development[xxix]. While much curriculum development is driven by content, the UBD framework involves what the authors call "backward design". That is, instead of choosing content and designing learning around it, UBD dictates first defining the objectives around what students should be able to understand for each topic and designing the curriculum (including content and assessment) around those. These objectives should be based on social needs or expectations around the subject area, learning standards, and research into the kind of understanding need for application of learning[xxx].

> Finally, the structure of the MCSG has been developed in a way that aims to build expertise in computer science, which is necessary if students wish to pursue further study or enter the workforce as computer scientists. Many curriculum and exam systems focus on students' ability to absorb information and simply recall it on an exam. This does not develop expertise, which involves knowing the conditions and contexts in which it is appropriate to apply certain parts of the knowledge one has acquired[xxxi]. Experts have a lot of knowledge but are able to retrieve and employ what they need for the specific scenario, something which high-stakes exams and other test can fail to assess.
>
> The design and progression of the MCSG, including Domains, Big Ideas, and Big Questions with application in real-world scenarios, aims to take all of this research into account.

Structure of the Curriculum

At every stage, the MCSG aims to address the aforementioned issues with current CS curricula, and to meet the needs expressed by governments. The structure of the curriculum and the chosen terminology are designed to address the issue of CS curricula being unappealing to learners. The MCSG makes use of descriptors that are more engaging for learners and lend themselves to project- or problem-based approaches to learning computer science. Specifically:

At the top organisational level of the MCSG are **Domains**. Domain replaces the term 'topic' in a traditional curriculum.

Underneath Domains are **Big Ideas**, which represent a learning pathway and continue throughout a student's journey through the MCSG, from age 5 to age 18. Big Ideas might be referred to as 'themes' in a traditional curriculum. In other words, they are concepts that serve as the main point around which a learning pathway is structured.

Each Big Idea contains **Big Questions**, which are akin to 'modules' in a traditional curriculum. A Big Question specifies the main subject of a discreet package of learning and contains its own outcomes.

Below each Big Question are learning goals, concepts and challenges, a link to a UN Sustainable Development Goal and specifications for supporting content.

The challenges lead to specific Computational Thinking, Statistical Literacy and Design Thinking competencies.

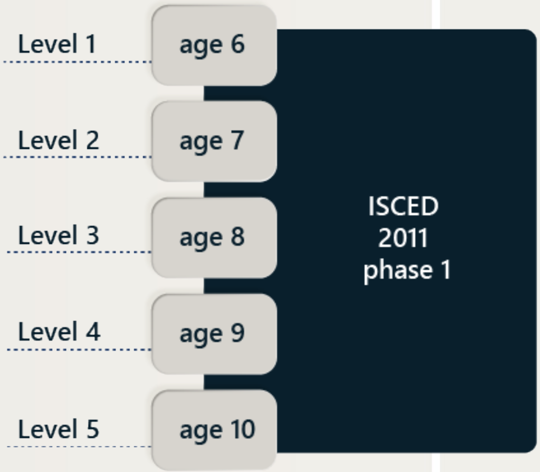
Phases

The MCSG follows the International Standard Classification of Education (ISCED 2011) created by UNESCO, which is used by international organisations to provide common terminology for education phases.[xxxii] It is described in Table 6.

Phase 1	Primary education
Phase 2	Lower secondary education
Phase 3	Upper secondary education
Phase 4	Post-secondary non-tertiary education

Table 6. ISCED phases and their equivalent school level.

These phases are mapped to ages, and each year is given a level. The age shown will be the maximum ages of students in a year group; for example, where Table 7 indicates Age 6, the age range for that level is ages 5 to 6. The levels map to the US "K" calibration.



Level 1	age 6	ISCED 2011 phase 1
Level 2	age 7	
Level 3	age 8	
Level 4	age 9	
Level 5	age 10	

Table 7. ISCED levels and their corresponding ages.

Table 8 sets out the key features of the curriculum as it applies to each phase. Students move though 4 levels of Design Thinking, lower to higher-order thinking skills[xxxiii]:

Table 8. Key features of the MCSG at each ISCED level.

ISCED 2011 Phase		Levels of thinking	Description – MCSG
1	Primary education	- **Foundation Level** - **Knowledge** - **Comprehension** - **Interpreting** - **Explaining** - **Identifying** - **Factual –** knowledge of terminology, specific details, and elements	Here, students should be provided with fundamental skills in CS and establish a solid foundation for learning. To enable students to follow clear learning pathways, the patterns of themes and topics established in Phase 1 are carried through to Phase 4. In most countries Phase 1 is less structured and rigidly timetabled as subsequent schooling phases, and the types of learning challenges presented to the students in this curriculum framework will reflect this.
2	Lower secondary education	- **Product Level** - **Apply** - **Implementing** - **Executing** - **Translating** - **Conceptual –** Knowledge of classification and categories, principles, and generalisations	Students in the first stage of secondary education will build on their primary CS education, following the same patterns and learning pathways established in Phase 1. The main difference between Phase 1 and Phase 2 is that CS will most likely be formally timetabled alongside other key subject areas.

3	Upper secondary education	• **Project Level** • **Analyse** • **De-constructing** • **Differentiating** • **Organising** • **Conceptual –** Knowledge of classification and categories, principles, and generalisations	In the second stage of secondary education the MCSG specifies learning activities which prepare students for tertiary education and/or acquiring skills relevant to employment. Usually in this phase students will pick subjects from a range of options, so CS is likely to be competing with other subjects for student enrolment. The key difference between this phase and the previous phases is that students who select CS will now have more time dedicated to the subject. This is reflected in the MCSG by an increase in the number of units that students would be expected to take in this phase.
4	Post-secondary non-tertiary education	• **Business Level** • **Synthesis** • **Coordinating** • **Critiquing** • **Testing** • **Procedural –** Application of specific skills, techniques, and methods	Here, the MCSG specifies learning experiences that build on Phase 3 and prepare students for the labour market; entry to tertiary education; or directly into entrepreneurship. Subject matter straddles upper secondary and tertiary education.

Compulsory CS education and the MCSG

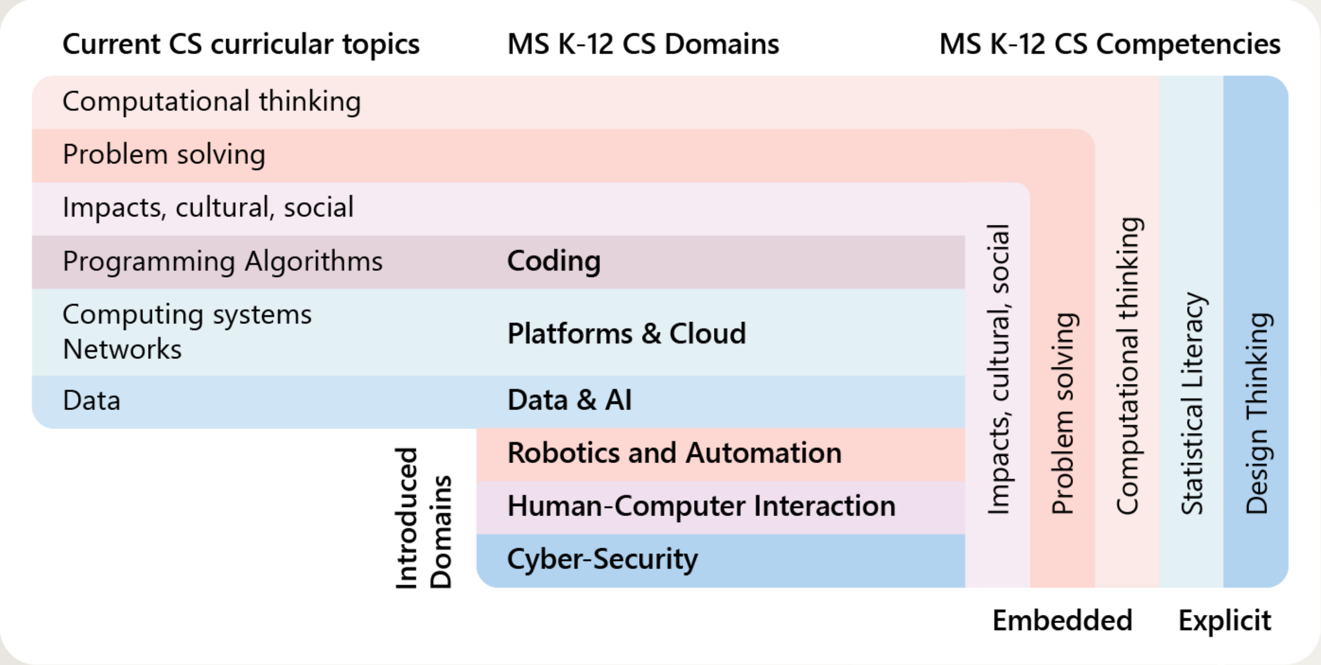
There is very little international consistency in how countries apply their national curricula across year groups in primary and secondary education. [xxxiv] For example, in some countries CS is an option offered only at the secondary level, whereas in others, it forms part of the national, required curriculum throughout a child's primary and secondary school education. The MCSG curriculum is based on the expectation that Phases 1 and 2 are compulsory; in other words, all students in primary and lower secondary are entitled to take CS, whilst Phases 3 and 4 are optional. The MSCG is designed so that up to Level 10 the CS curriculum will be taken by all children.

Between Levels 10 and 13, students elect to take the subject, so more time is spent in the curriculum compared to previous years.

Domains

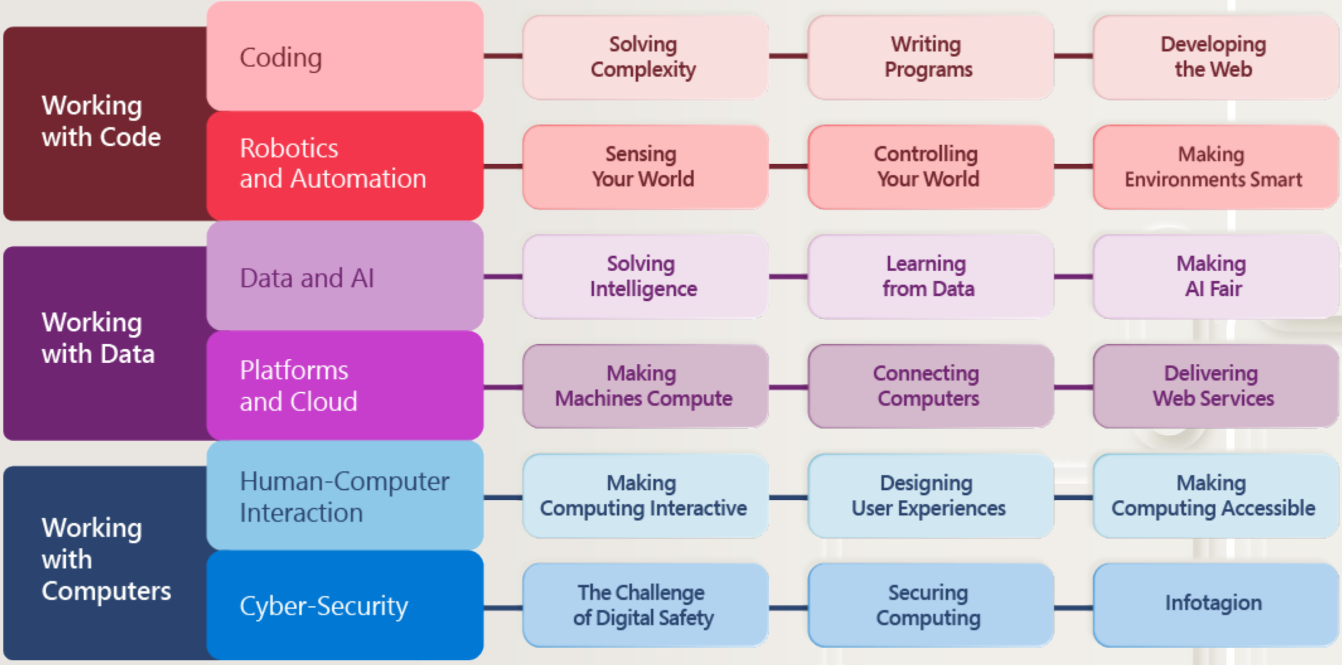
This curriculum specifies six Domains. Three of these Domains align with established computer science curriculum topics. Three new Domains have been established to address currency and appeal problems with existing curricula. Each of these Domains and what they contain are explained in detail in section X.

Figure 1 shows how existing CS curricular topics (see Table 3) have been integrated into the curriculum, where new Domains have been created, and competencies added.



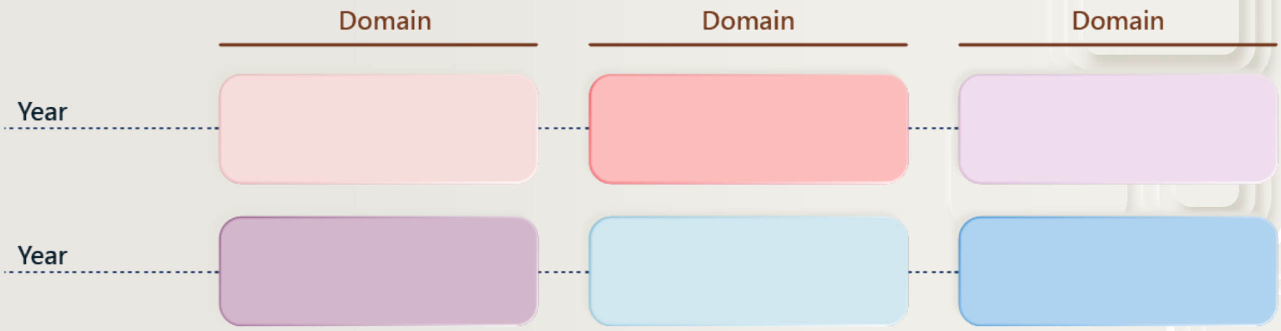
Domains in Primary Education

Whilst each of the six domains listed in Table 2 are also present in Phase 1 of the MCSG, the terminology in the domain titles might seem daunting or overly complex for learners in primary school. For this reason, the six domains have been organised under three over-arching categories that are more suitable to primary school teachers, learners, and parents. Those three categories are working with code, working with data, and working with computers. The domain organisation underneath those categories is represented as follows.



Working with Code	Coding	Solving Complexity	Writing Programs	Developing the Web
	Robotics and Automation	Sensing Your World	Controlling Your World	Making Environments Smart
Working with Data	Data and AI	Solving Intelligence	Learning from Data	Making AI Fair
	Platforms and Cloud	Making Machines Compute	Connecting Computers	Delivering Web Services
Working with Computers	Human-Computer Interaction	Making Computing Interactive	Designing User Experiences	Making Computing Accessible
	Cyber-Security	The Challenge of Digital Safety	Securing Computing	Infotagion

Most countries operate between 2 and 5 academic terms[xxxv]. Therefore three 'Domains' are specified per year. Each year, students will complete a set of learning objectives for three Domains.

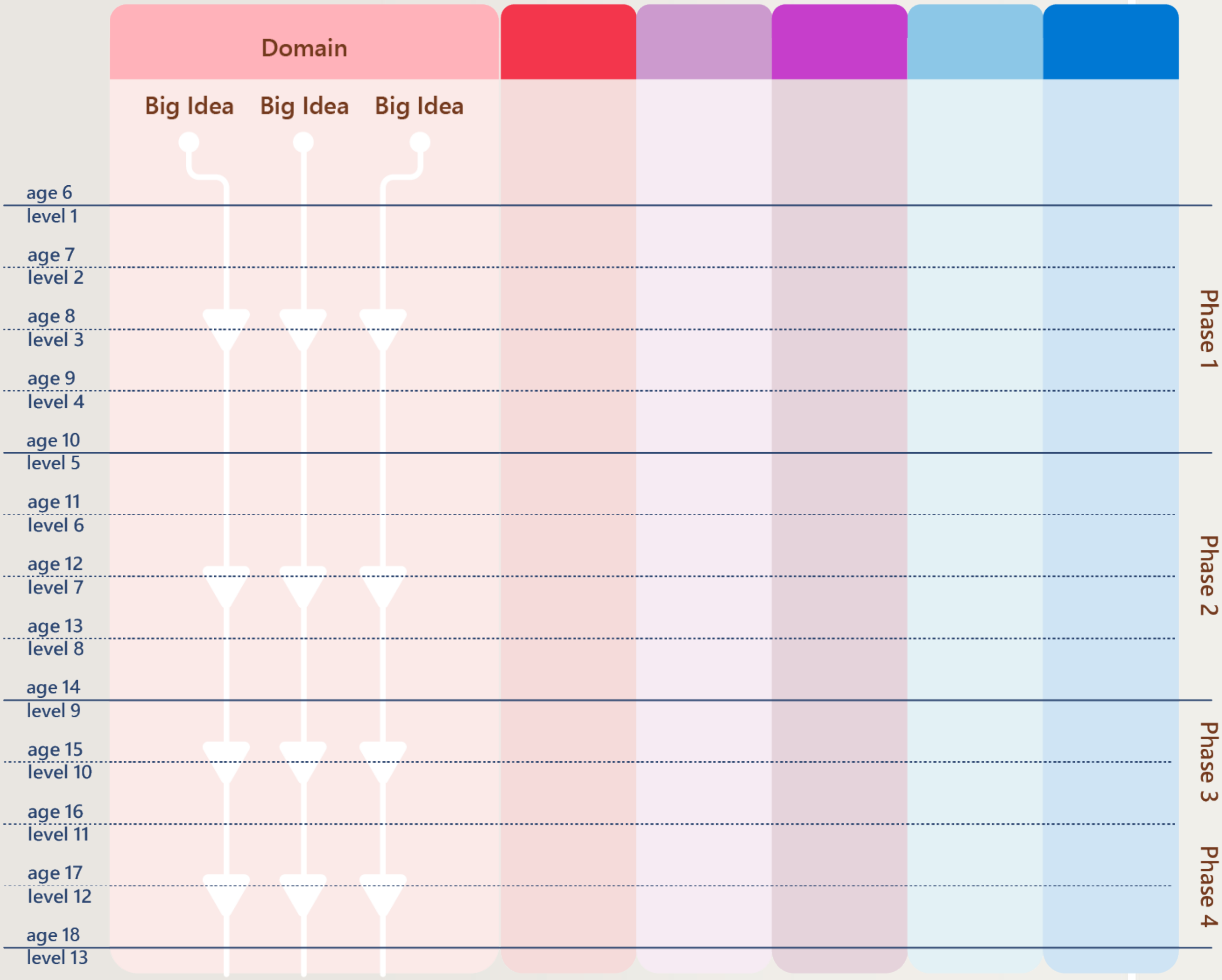


As there are six Domains in each Phase, three Domains are covered in one year, and the other three Domains are covered the following year.

This process repeats until Level 10.

Big Ideas

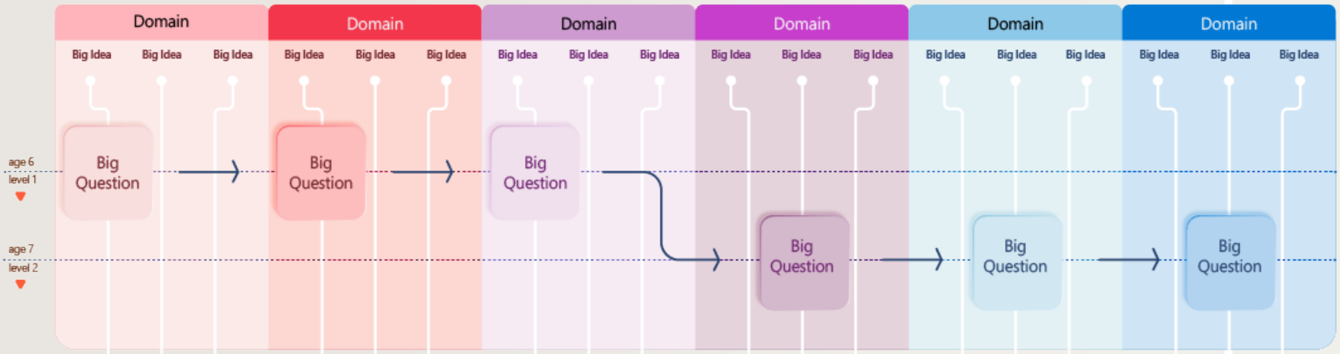
Each Domain contains three Big Ideas which contain learning pathways. An important feature of this curriculum framework is that learning pathways – Big Ideas – extend from Phase 1 to Phase 4.



Big Questions

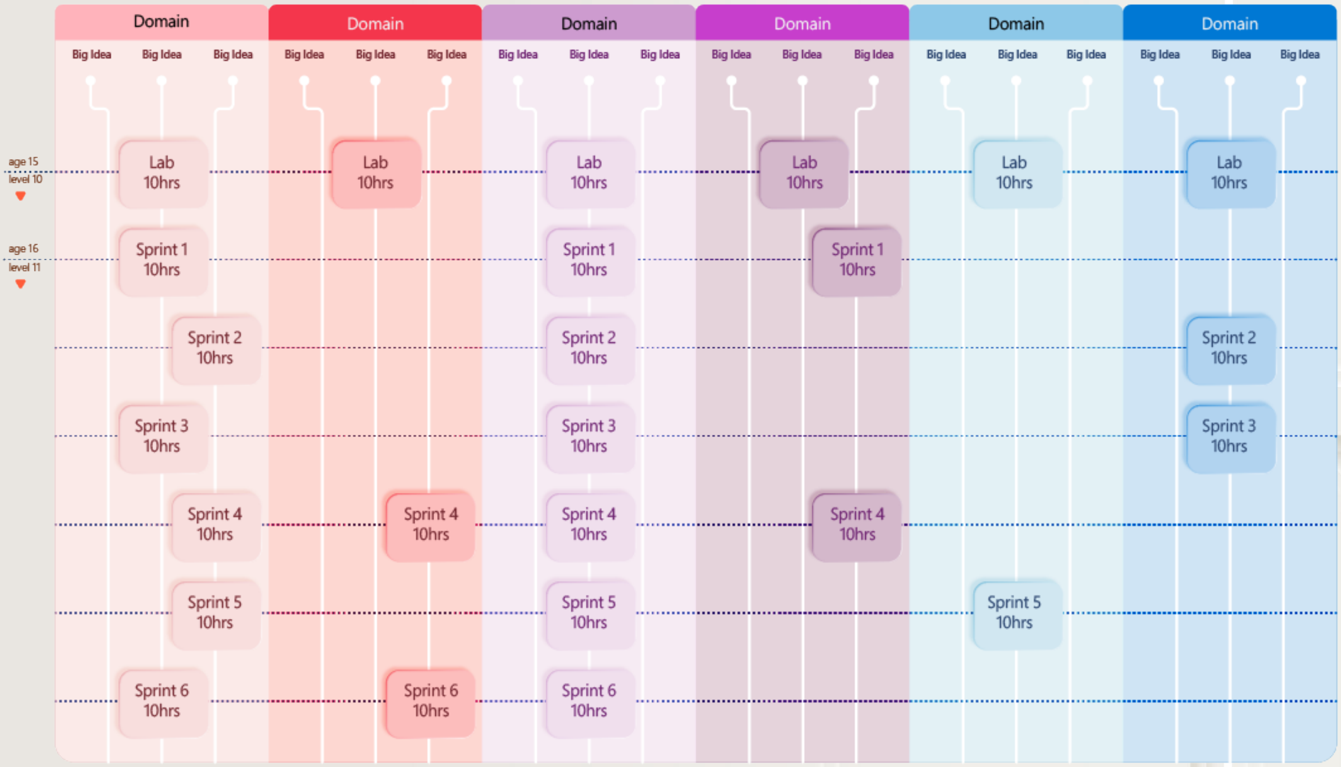
Each Big Idea contains Big Questions. In Phases 1 and 2, students will work through two or three Big Questions per Domain. The graphic below shows the placement of the Big Questions in the darker boxes.

Chronological progression through the curriculum involves students working on one Big Question in one Domain at a time.

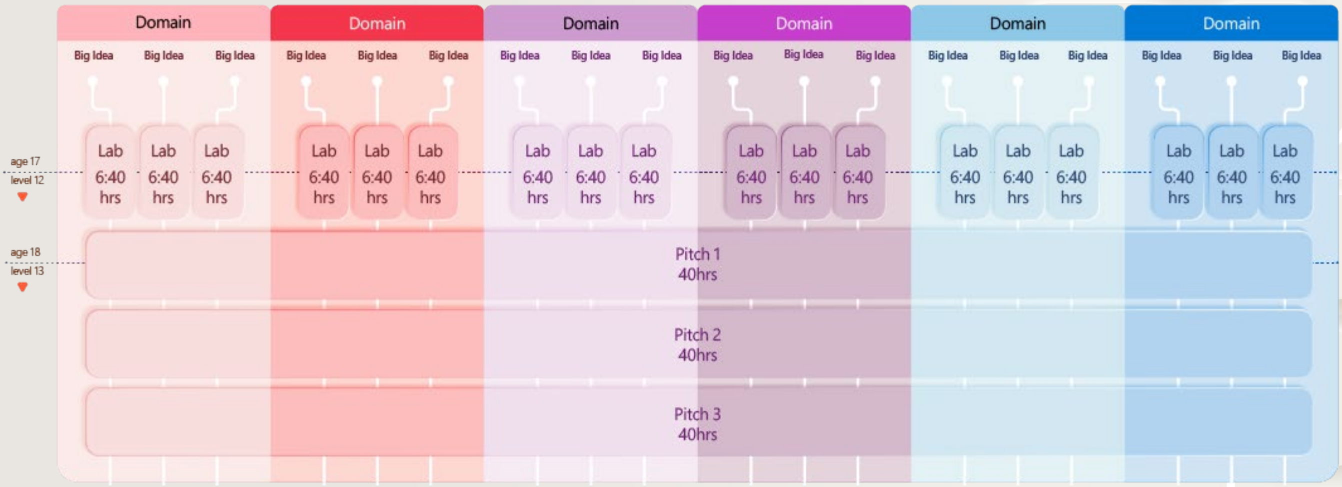


Three Big Questions are specified per year for Phases 1 and 2.

In Phase 3 (based on UK GCSE) year 1, the MCSG recommends six 'labs' followed by six short open-ended projects in year two (one per half-term using the UK as a template). The Big Question of the lab is similar to what students would do in science and engineering subjects at University – i.e. experiments or practical activities with a specific learning goal.



In Phase 4 (based on UK A 'Level) the MCSG specifies 18 'labs' for year 1, followed by three longer-term open-ended projects in the final year.



Time allocation

To develop the following time allocation recommendations for each phase of the curriculum, the authors consulted guidelines provided by the two United Kingdom exam boards (OCR and AQA), the International Baccalaureate computer science curriculum, and multiple schemes of work provided to primary and secondary school teachers on the Computing at School online community[xxxvi]. These guidelines are to aid teachers in their planning for implementing this curriculum; it is possible that the actual time required may differ slightly or may need to be adjusted to suit the requirements of a local education system.

Phase	1-year provision	Time each	Total Time
1	**3 Big Questions**	10 hours	30 hours
2	**3 Big Questions**	10 hours	30 hours
3 yr. 1	**6 'Labs'**	10 hours	60 hours
yr. 2	**6 Projects (Sprints)**	10 hours	60 hours
4 yr. 1	**18 'Labs'**	6:40 hours	120 hours
yr. 2	**3 Projects (Pitches)**	40 hours	120 hours

Learning Goals

Each Big Question sets out learning goals. Learning goals will begin with active verbs that refer to cognitive processes set out in k12cs.org's K-12 Computer Science Framework[xxxvii], including, for example –

- Understand
- Appreciate
- Know

Concepts

Each Big Question specifies what key concepts should be learned.

Example Lesson Outlines

The lesson outline suggestions provided here are intended to sketch a learning pathway for teachers to use in their detailed lesson planning. Each unit is based on an hour lesson time, and has links to supporting resources.

Lessons should be practical and where possible contains challenges. The lesson outlines are designed to produce clear, assessable outcomes and enable students to acquire competencies in computational thinking, data literacy and design thinking, as discussed in the following sections.

Computational Thinking

Andreas Schleicher, Director of the OECD Directorate for Education and Skills recommends focusing on the computational thinking skills that students can use to shape the technologies of tomorrowxxxviii. Key abilities include:

- *Decomposition* – breaking down a complex problem or system into smaller, more manageable parts.
- *Pattern recognition* – looking for similarities among and within problems
- *Abstraction* – focusing on the important information only, ignoring irrelevant detail
- *Algorithms* – developing a step-by-step solution to the problem, or the rules to follow to solve the problem

Data Literacy

Data Literacy is the ability to read, work with, analyse, communicate about, and question data and statistics. It is an essential skill for understanding and being able to explain the workings of AI, for example.

Design Thinking

Developed at the Stanford d.school[xxxix], Design Thinking is a methodology that teaches individuals new strategies to solve problems. The design process challenges students to combine empathy, ingenuity and rationality to meet user needs and create successful solutions with an innovator's mindset. Students are taught to defer judgment early in the process, which reduces fear of failure and encourages thinking outside the box.

A core innovation skillset, design thinking goes beyond problem solving. Design thinking is an iterative approach that uses prototyping for continuous user feedback and engagement and is an essential skill for developing products – a key goal of the MCSG.

Design Thinking is also a practical tool for integrating 21st century skills into the classroom. It makes direct connections between content students learn in class and the world beyond their school. Key steps in the process include:

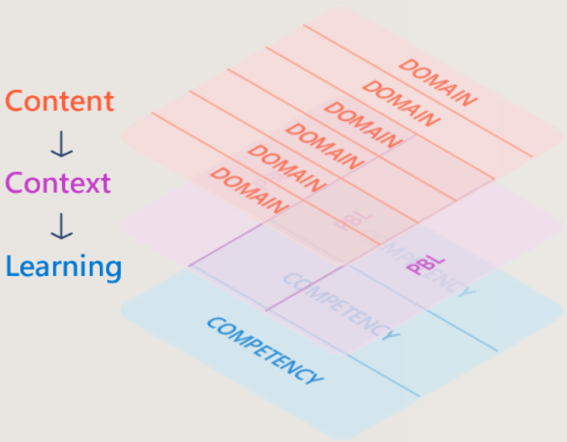
- Empathise
- Define
- Big Ideate
- Prototype
- Test

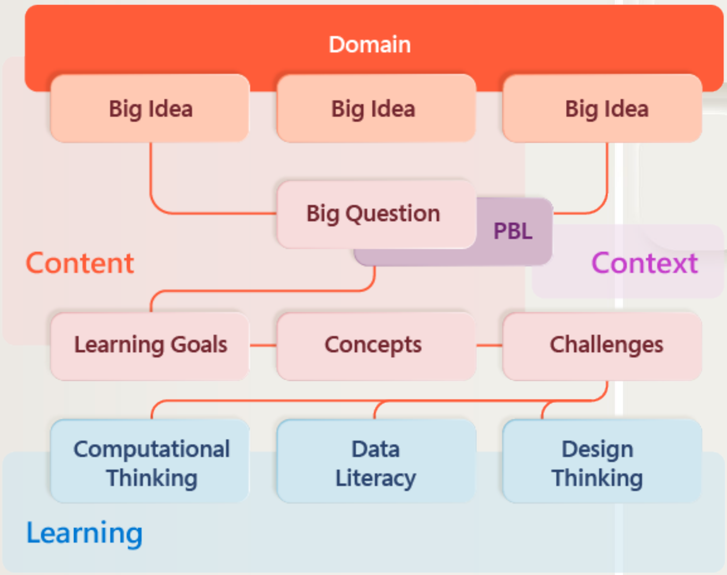
Supporting Content

Each Big Question includes recommendations for content to support the challenges, which originate from a wide range of sources. Local content can easily be used to replace or augment the examples provided.

Structure Summary

At the highest level, the curriculum framework structure is depicted in the following figures:





Context is provided through project-based learning approaches at the Big Question level. Content is organised at Domain levels, where each Domain contains three Big Ideas under which are the Big Questions. The end goal is the learning, which is defined by the acquisition of three competencies – Computational Thinking, Data Literacy, and Design Thinking.

Attainment Targets and Progression

The MCSG attainment targets and progressions have two elements:

- Competencies
- Transversal skills and knowledge

Competencies

The MCSG is aligned to "Computing Progression Pathways, CASxl" and "Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017xli".

However, MCSG goes beyond these standards with additional innovative thinking that reflects the need to emphasise areas such as AI, The Cloud, IoT, and emerging human-computer interfaces; and to encourage creativity and entrepreneurialism.

Therefore, the MCSG explicitly prescribes the following sets of competencies, as mentioned previously: computational thinking, data literacy, and design thinking. The standards associated with each set of competencies in the MCSG are described in the following sections.

Computational Thinking

The computational thinking descriptors are based on two sets of standards:

- Computing Progression Teachers Association (CSTA) K-12[xlii] – labelled "CTSA" in the following tables.
- Computer Science Standards[xliii] – marked in blue, labelled "CAS" in the following tables.

Other descriptors are bespoke to the MCSG, and are marked in green and labelled "MCSG" in the following tables.

By the end of Phase 1 (Foundation Level), Students Can:

Level (K)	Competency Descriptor	Standard	Ref
1	Understand fundamentals of what a computer is – hardware and software working together to process data	MCSG	
1	Explain that computers collect data from various input devices, including sensors and application software.	CAS	H&P 3
1	Understand that computers have no intelligence and that computers can do nothing unless a program is executed.	CAS	H&P 1
2	Explain the role of electricity in computing systems and processes	MCSG	
3	Write sequences, events, loops, and conditionals	MCSG	
3	Create programs that include sequences, events, loops, and conditionals	CSTA	1A-AP-10
3	Use post-tested loops e.g. 'until', and a sequence of selection statements in programs, including an 'if', 'then' and 'else' statement	CAS	P&D 3
3	Program robots	MCSG	
4	Make a cipher work	MCSG	
5	Construct static web pages using HTML and CSS	CAS	C&N 5

By the end of Phase 2 (Product Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
7	Manage complexity using established methods, procedures and tools for structuring and managing projects; and producing, versioning and sharing code	MCSG	
7	Use C code functions, variables, and structures in a control solution	MCSG	
7	Apply practical experience of a high-level textual language	CAS	P&D 5

8	Describe how internal and external parts of computing devices function to form a system	CSTA	1B-CS-01
8	Understand the main functions of the operating system	CAS	H&P 4
8	Explain and use strong passwords to protect devices and information from unauthorized access	CSTA	1A-NI-04
8	Discuss real-world cyber-security problems and how personal information can be protected	CSTA	1B-NI-05
8	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts	CSTA	3A-NI-06
9	Develop programs that combine control structures, including nested loops and compound conditionals	CSTA	2-AP-12

By the end of Phase 3 (Project Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
10	Use data analysis tools and techniques to identify patterns in data representing complex systems	CSTA	3B-DA-05
10	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects	CSTA	3A-AP-17
10	Access pre-existing functionality from standard libraries	MCSG	
10	Program an RPA process	MCSG	
10	Know the relative merits of different network methods, the role of Internet protocols and how packets, IP addresses, and memory work	MCSG	
10	Understand data transmission between digital computers over networks, including the internet i.e. IP addresses and packet switching	CAS	C&N 5
10	Explain the names of hardware e.g. hubs, routers, switches, and the names of protocols e.g. SMTP, iMAP, POP, FTP, TCP/ IP, associated with networking computer systems.	CAS	C&N 5
10	Explain security issues that may lead to compromised systems	CSTA	3B-AP-18
10	Apply multiple methods of encryption to model the secure transmission of information.	CSTA	2-NI-06
11	Apply prior descriptors to each of the six Sprints		

By the end of Phase 4 (Business Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
12	Understand the power of OOP for modelling real world phenomena, create a DRY OO program with high modularity and extensibility	MCSG	
12	Apply JavaScript to create responsive, interactive webpages	MCSG	
12	Describe how artificial intelligence drives many software and physical systems	CSTA	3B-AP-08
12	Integrate different types of sensors in an autonomous system	MCSG	
12	Apply a range of mechatronic skills to solve robotics problems	MCSG	
12	Apply a range of IoT skills to solve smart city/smart environment problems	MCSG	
12	Write a Classical Machine Learning algorithm to classify Earth images	MCSG	
12	Demonstrate an understanding of number systems, computing electronics, memory, ICs, processor types, fetch-execute cycle	MCSG	
12	Parameterise a Neural Network to solve a predictive modelling problem	MCSG	
12	Write code to ingest data from a public API	MCSG	
12	Explain the fundamentals of Quantum computing	MCSG	
13	Application of prior descriptors to each of the three Pitches		

Data Literacy

Data Literacy is the ability to understand and reason with statistics and data. It is an essential skill for understanding and being able to explain the workings of AI. These competencies also incorporate aspects of the Australian ACARA Digital Technologies Curriculumxliv and Numeracy Progressionsxlv as well as the CAS and CSTA standards described previously.

By the end of Phase 1 (Foundation Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
1	Collect, visualise and explain patterns in data	MCSG	
1	Recognise that data can be structured in tables to make it useful	CAS	D&DR 1
1	Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data	CSTA	1A-DA-05
1	Identify and describe patterns in data visualizations, such as charts or graphs, to make a prediction	CSTA	1A-DA-07
1	Demonstrate an understanding of the basic concepts of human intelligence	MCSG	
2	Know that digital computers use binary to represent all data, how it can represent numbers and images, how computers transfer data in binary relationship between binary and file size	CAS	D&DR 5
3	Identify your position and calculate the position of a landmark or object relative to your position in the real-world	MCSG	
3	Express probabilities numerically	MCSG	
3	Understand that data can be learned from	MCSG	
5	Explain how IoT can be combined with Cloud Computing to deliver aggregated data from around the world	MCSG	
5	Explain the concepts of bias and fairness in the context of AI and automation	MCSG	
5	Understand that personal data is collected, and it is important to keep this information secure	MCSG	
5	Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access	CSTA	1A-NI-04
5	Keep login information private, and log off of devices appropriately	CSTA	1A-1C-18

By the end of Phase 2 (Product Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
6	Use units and scales of computing	MCSG	
7	Process Serial Data in a control solution	MCSG	
7	Understand how data can be used in the creative world	MCSG	
9	Create interactive data visualizations using software tools to help others better understand real-world phenomena	CTSA	3A-DA-11
9	Develop, apply, interpret and communicate statistical models, judgements and arguments	MCSG	
9	Use algorithms to make predictions by turning probability concepts into code	MCSG	
9	Apply combined control technologies in a Cloud-based IoT solution	MCSG	
9	Develop, apply, interpret and communicate statistical models, judgements and arguments	MCSG	
9	Explain how data is encoded from a range of media, how it is used to make predictions, and the effects of bias	MCSG	
9	Discuss issues of bias and accessibility in the design of existing technologies	CSTA	2-IC-21
9	Query data on one table using a typical query language	CAS	D&DR 5

By the end of Phase 3 (Project Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
10	Create a computational model that represents the relationships among different elements of data collected from a phenomenon or process.	CSTA	3A-DA-12
10	Evaluate quality, authenticity and accuracy of data and extrapolate from a trend or pattern. Use a range of models and charting methods to analyse, predict and communicate data stories	MCSG	
10	Clean and prepare textual data for analysis and Machine Learning	MCSG	
10	Understand the role of mathematics in encryption	MCSG	
11	Apply prior descriptors to each of the six Sprints		

By the end of Phase 4 (Business Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
12	Explain what a relational database is, and understand the benefits of storing data in multiple tables	CAS	D&DR 8
12	Query, join, transform and summarize data into useful information using a typical query language such as SQL	MCSG	
12	Demonstrate understanding of the principles of Deep Learning and its power, potential and limitations	MCSG	
12	Apply supervised learning methods to classify image data	MCSG	
13	Apply prior descriptors to each of the three Pitches		

Design Thinking

Design Thinking is the ability to combine empathy, ingenuity and rationality to meet user needs to solve problems. It is an essential creative skill for being able to develop solutions to problems and products. These competencies incorporate aspects of the CAS and CSTA standards described above.

By the end of Phase 1 (Foundation Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
2	Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware)	CSTA	1A-CS-02
2	Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination	CSTA	1B-NI-04
2	Explain relative merits of different types of interface	MCSG	
2	Seek diverse perspectives for the purpose of improving computational artefacts	CSTA	1B-IC-20
2	Demonstrate safe and responsible computing	MCSG	
2	Compare how people live and work before and after the implementation or adoption of new computing technology	CSTA	1A-1C-16
4	Think of ways to improve the accessibility and usability of computing	MCSG	
4	Evaluate the trustworthiness of digital content and consider the usability of visual design features when designing and creating digital artefacts for a known audience	CAS	IT 6

By the end of Phase 2 (Product Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
6	Arrange elements of a Cloud-based stack, including a database, in a diagram	MCSG	
6	Recognise and understands the function of the main internal parts of basic computer architecture	CAS	H&P 5
6	Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users	CSTA	3A-AP-19
6	Recommend improvements to design of computing experiences	MCSG	
6	Establish ethical protocols for the online world	MCSG	
6	Compare trade-offs associated with computing technologies that affect people's everyday activities and career options	CSTA	2-IC-20
6	Describe trade-offs between allowing information to be public and keeping information private and secure	CSTA	2-IC-23
7	Manage complexity with diagrams, procedures and tools and know how to organize, version, share and reuse code	MCSG	
8	Use reverse engineering to understand a solution	MCSG	

By the end of Phase 3 (Project Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
10	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices	CSTA	2-CS-01
10	Apply principles of UX design	MCSG	
10	Understand the importance of fail-safe and zero errors in safety-critical systems	MCSG	
11	Produce use case scenarios, plan sequences, goals, and outcomes	MCSG	
11	Combine hardware, software and network technologies	MCSG	
11	Modify, remix or incorporate parts of an existing program	MCSG	
11	Apply Design Thinking – Empathise, Define, Ideate, Prototype, Test	MCSG	

By the end of Phase 4 (Business Level), students can:

Level (K)	Competency Descriptor	Standard	Ref
12	Apply project management and sustainability principles to apply software to a sustainability problem	MCSG	
12	Apply Business Model Canvas	MCSG	
12	Collaboratively deploy and manage software artefacts on a code management platform	MCSG	
12	Discuss the "five tribes of AI"	MCSG	
12	Demonstrate a working understanding of XAI and AI Ethics	MCSG	
12	Demonstrate a working understanding of the criticality of factoring energy use into a solution	MCSG	
12	Demonstrate a working understanding of haptics, tracking and other advanced HCI technology	MCSG	
12	Use User-centric Design methods	MCSG	
12	Factor-in accessibility when designing solutions	MCSG	
12	Design smart contract processes	MCSG	
12	Demonstrate an appreciation of technology law	MCSG	
13	Apply sound fundamentals of system organisation, architecture, and solution design	MCSG	
13	Collaborate effectively in a software team using version control and continuous deployment for commerce-ready software	MCSG	
13	Effectively use prototyping	MCSG	
13	Pitch convincingly to investors	MCSG	

Transversal Skills and Knowledge

Cutting across all Domains, Big Ideas and Big Questions are the following transversal skills and knowledge:

Digital Literacy

By the time students are 7 years of age (level 2), they should be able to use technology purposefully and safely to create, organise, store, manipulate and retrieve digital content[xlvi].

By age 10 (level 5) students should be able to use basic cloud services independently.

The Microsoft Digital Literacy Course (MDL) offers structured pathway to enable this. The table below shows how it maps to the MCSG Big Questions.

Level (K)	Big Question	MDL unit	Description
2	**What does 'digital' mean?**	1 & 2	Work with Computers Access information online
2	**How can computers and people interact?**	3 & 5	Communicate online Create digital content
2	**How Can You Stay Safe Online?**	4	Participate Safely and Responsibly Online
5	**How can we connect sensors across schools?**	6	Collaborate and Manage Content Digitally

Completing the MDL course by the end of the Foundation Phase (Phase 1), means that students will have the key basic digital literacy skills that they need to be successful in the MCSG and across the rest of their schooling.

Information Technology

Information Technology (IT) focusses on using computers whilst the focus of CS is on the science of computing and being able to be create computing solutions. CS requires extensive use of IT, so the use of a wide range of IT tools and services is embedded within the MCSG, including those listed in the following table.

Phase	IT Tools and services
Foundation	Windows, Office, PDF, MakeCode, Minecraft Education, OneDrive, sensors, Azure
Product	Excel, Visio, Project, Arduino IDE, MR interfaces, Raspberry Pi, Azure Notebooks, Visual Studio
Project	IDLE, RPA, bots, network management technologies, encryption
Business	GitHub, UML, SQL, Azure Machine Learning, Blockchain, Haptics

Future-Ready Skills

Skills relating to cloud computing, artificial intelligence, machine learning, productivity and more are already in demand in organizations around the world, yet jobs remain unfilled, and the talent gap persists and is set to widen. More than half of today's jobs require technology skills, but in less than a decade that number will grow to more than 77%[xlvii].

The Microsoft Future-Ready Skills programme provides a framework to support students and educators from K-Career to be ready to innovate and create in an increasingly digital world. The following table shows how the MCSG maps directly to the Microsoft Future-Ready Skills programme.

Phase	Foundation	Product	Project	Business
Future-ready Skill	Block-based visual programming	Text-based programming		
	Digital Literacy	Productivity		
		Cloud		
	Modern skills – communication, collaboration, creativity, critical and computational thinking			
	Computer Science Fundamentals			
	Programming and design thinking			
	Technology ethics and data concepts			

Troubleshooting

Core attributes required by students learning how to create technology as future practitioners of CS are resilience and the ability to troubleshoot.

The MCSG also maps to the development of troubleshooting skills as described by the Computer Science Teachers Association (CSTA) standards from the United States as indicated in the following table.

Phase	Troubleshooting Skill	CSTA Standard
Foundation	Describe basic hardware and software problems using accurate terminology	1A-CS-03
Product	Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies	1B-CS-03
Project	Systematically identify and fix problems with computing devices and their components	2-CS-03
Business	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors	3A-CS-03

Impacts of Computing

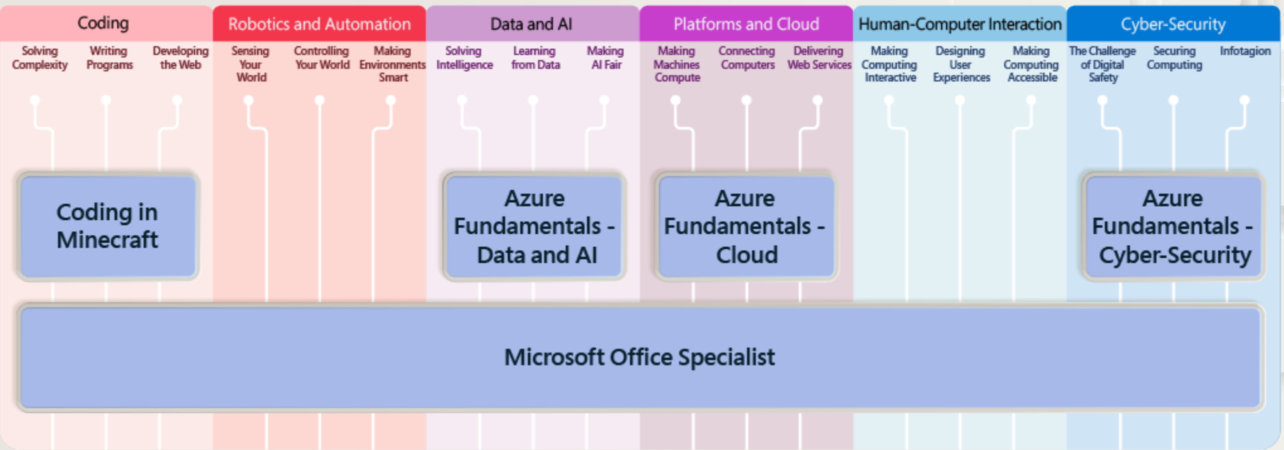
In addition to the technical and career-focused skills that the MCSG aims to develop, it's also essential that students understand the impacts of computing on both themselves and other people, and develop awareness of what counts as acceptable or otherwise.

The development of the awareness of the impacts of computing, as described by CSTA, is depicted in the following table.

Phase	Impact of Computing Knowledge and Skills	CSTA
Foundation	Compare how people live and work before and after the implementation or adoption of new computing technology	1A-IC-16
	Work respectfully and responsibly with others online	1A-IC-17
	Keep login information private, and log-off devices appropriately	1A-IC-18
Product	Discuss computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices	1B-IC-18
	Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users	1B-IC-19
	Seek diverse perspectives for the purpose of improving computational artefacts	1B-IC-20
	Observe intellectual property rights and give appropriate attribution when creating or remixing programs	1B-AP-14
Project	Compare trade-offs associated with computing technologies that affect people's everyday activities and career options	2-IC-20
	Discuss issues of bias and accessibility in the design of existing technologies	2-IC-21
	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artefact	2-IC-22
	Describe trade-offs between allowing information to be public and keeping information private and secure	2-IC-23
Business	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices	3A-IC-24
	Test and refine computational artefacts to reduce bias and equity deficits	3A-IC-25
	Demonstrate ways a given algorithm applies to problems across disciplines	3A-IC-26
	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields	3A-IC-27
	Explain the beneficial and harmful effects that intellectual property laws can have on innovation	3A-IC-28
	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users	3A-IC-29
	Evaluate the social and economic implications of privacy in the context of safety, law or ethics	3A-IC-30

Certification Pathways

Whilst schools will have their own methods for formative and summative assessment, certification needs to be considered and planned for. Microsoft certification in primary and secondary schooling builds towards professional level exams in the following way:



A good place to start with certifications is for all students to gain Microsoft Office skills and to get these skills recognised through certification. Using Microsoft Office is the backbone of vast areas of modern work and gaining the foundational skills and confidence in Microsoft Office sets up students for the world of work[xlviii].

There are opportunities to gain and certify these skills throughout the entire Microsoft Computer Science Guide, at all levels of schooling and in all domains through the Microsoft Office Specialist (MOS) program. As students progress through the curriculum, they will need to produce work using each of the Office products that are certified in MOS – e.g., Excel for working with data; PowerPoint for making presentations; Word for producing reports; and Outlook for communicating and collaborating. The ideal time for students to gain MOS certification would be at the end of the 'Product' level in the MCSG framework.

For students aged between 11-15 years, Coding in Minecraft provides a pathway of introductory courses and assessments to develop and prove their coding and introductory computer science skills using MakeCode and JavaScript or Python[xlix]. Exam Objective Domains are[l] –

- Designing & Developing Algorithms
- Block-based coding using MakeCode
- Text-based coding

For students aged between 16-18 years – certification pathways can include the following Microsoft technical "Fundamentals" certification supported by "Pre-Fundamentals" learning content:

Azure Cloud Fundamentals, Exam AZ-900: Microsoft Azure Fundamentals – Certifications | Microsoft Learn

- Pre-Fundamentals: Describe cloud computing – Training | Microsoft Learn

Azure AI Fundamentals, Exam AI-900: Microsoft Azure AI Fundamentals – Certifications | Microsoft Learn

- Pre-Fundamentals: Get started with AI on Azure – Training | Microsoft Learn

Azure Data Fundamentals, Exam DP-900: Microsoft Azure Data Fundamentals – Certifications | Microsoft Learn

- Pre-Fundamentals: Understand data concepts – Training | Microsoft Learn

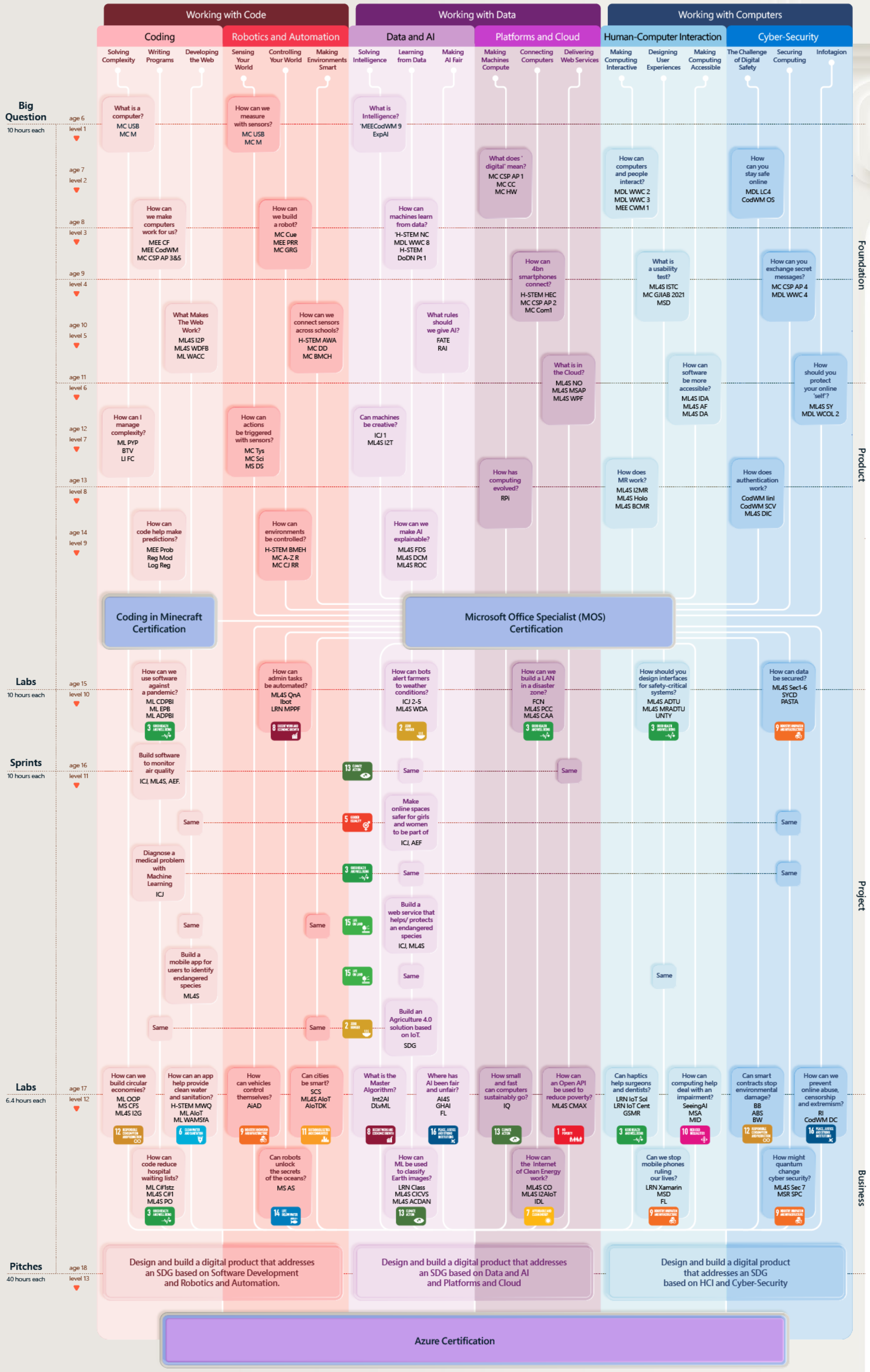
Azure Cybersecurity Fundamentals, Exam SC-900: Microsoft Security, Compliance, and Identity Fundamentals – Certifications | Microsoft Learn

- Pre-Fundamentals:
- Describe the basic concepts of cybersecurity – Training | Microsoft Learn
- Establish the guiding principles and core components of Zero Trust – Training | Microsoft Learn
- Examine ransomware and extortion-based threats – Training | Microsoft Learn

Learning pathways with Microsoft Official Coursework preparatory for these industry recognized Fundamentals certifications are available on Microsoft Learn. Each certificate learning pathway is also supported by "Pre-Fundamentals" learning content. For educators interested in incorporating official curricula for Fundamentals coursework and industry certification exams into their programs of study in school, the Microsoft Learn for Educators program provides instructor resources, including course data sheets, educator teaching guides, educator certifications, presentation materials, and lab exercises.

Framework Overview

This section describes the different perspectives that frame the curriculum in this guide. It explains the subjects covered in each of the Big Ideas, as well as the Big Questions that each Big Idea contains. The following figures depict the overview, and then gives a chronological view through MCSG. From there, there is a detailed explanation of the six Domains, each of which contain three learning pathways – the Big Ideas – and the Big Questions contained within them.


Working with Code
Working with Data
Working with Computers
Coding
Robotics and Automation
Data and AI
Platforms and Cloud
Human-Computer Interaction
Cyber-Security
Solving Complexity
Writing Programs
Developing the Web
Sensing Your World
Controlling Your World
Making Environments Smart
Solving Intelligence
Learning from Data
Making AI Fair
Making Machines Compute
Connecting Computers
Delivering Web Services
Making Computing Interactive
Designing User Experiences
Making Computing Accessible
The Challenge of Digital Safety
Securing Computing
Infotagion
Big Question
10 hours each
age 6
level 1
age 7
level 2
age 8
level 3
age 9
level 4
age 10
level 5
age 11
level 6
age 12
level 7
age 13
level 8
age 14
level 9
What is a computer? MC USB MC M
How can we measure with sensors? MC USB MC M
What is Intelligence? 'MEECodWM 9 ExpAI
What does ' digital' mean? MC CSP AP 1 MC CC MC HW
How can computers and people interact? MDL WWC 2 MDL WWC 3 MEE CWM 1
How can you stay safe online MDL LC4 CodWM OS
How can we make computers work for us? MEE CF MEE CodWM MC CSP AP 3&5
How can we build a robot? MC Cue MEE PRR MC GRG
How can machines learn from data? 'H-STEM NC MDL WWC 8 H-STEM DoDN Pt 1
How can 4bn smartphones connect? H-STEM HEC MC CSP AP 2 MC Com1
What is a usability test? ML4S ISTC MC GJIAB 2021 MSD
How can you exchange secret messages? MC CSP AP 4 MDL WWC 4
What Makes The Web Work? ML4S I2P ML4S WDFB ML WACC
How can we connect sensors across schools? H-STEM AWA MC DD MC BMCH
What rules should we give AI? FATE RAI
What is in the Cloud? ML4S NO ML4S MSAP ML4S WPF
How can software be more accessible? ML4S IDA ML4S AF ML4S DA
How should you protect your online 'self'? ML4S SY MDL WCOL 2
How can I manage complexity? ML PYP BTV LI FC
How can actions be triggered with sensors? MC Tys MC Sci MS DS
Can machines be creative? ICJ 1 ML4S I2T
How has computing evolved? RPi
How does MR work? ML4S I2MR ML4S Holo ML4S BCMR
How does authentication work? CodWM linl CodWM SCV ML4S DIC
How can code help make predictions? MEE Prob Reg Mod Log Reg
How can environments be controlled? H-STEM BMEH MC A-Z R MC CJ RR
How can we make AI explainable? ML4S FDS ML4S DCM ML4S ROC
Foundation
Product
Coding in Minecraft Certification
Microsoft Office Specialist (MOS) Certification
Labs
10 hours each
age 15
level 10
How can we use software against a pandemic? ML CDPBI ML EPB ML ADPBI
3
How can admin tasks be automated? ML4S QnA Ibot LRN MPPF
8
How can bots alert farmers to weather conditions? ICJ 2-5 ML4S WDA
2
How can we build a LAN in a disaster zone? FCN ML4S PCC ML4S CAA
3
How should you design interfaces for safety-critical systems? ML4S ADTU ML4S MRADTU UNTY
3
How can data be secured? ML4S Sec1-6 SYCD PASTA
9
Sprints
10 hours each
age 16
level 11
Build software to monitor air quality ICJ, ML4S, AEF.
13
Same
Same
Same
5
Make online spaces safer for girls and women to be part of ICJ, AEF
Same
Diagnose a medical problem with Machine Learning ICJ
3
Same
Same
Same
Same
15
Build a web service that helps/ protects an endangered species ICJ, ML4S
Build a mobile app for users to identify endangered species ML4S
15
Same
Same
Same
Same
2
Build an Agriculture 4.0 solution based on IoT. SDG
Project
Labs
6.4 hours each
age 17
level 12
How can we build circular economies? ML OOP MS CFS ML4S I2G
12
How can an app help provide clean water and sanitation? H-STEM MWQ ML AIoT ML WAMSfA
6
How can vehicles control themselves? AiAD
9
Can cities be smart? SCS ML4S AIoT AIoTDK
11
What is the Master Algorithm? Int2AI DLvML
8
Where has AI been fair and unfair? AI4S GHAI FL
16
How small and fast can computers sustainably go? IQ
13
How can an Open API be used to reduce poverty? ML4S CMAX
1
Can haptics help surgeons and dentists? LRN IoT Sol LRN IoT Cent GSMR
3
How can computing help deal with an impairment? SeeingAI MSA MID
10
Can smart contracts stop environmental damage? BB ABS BW
12
How can we prevent online abuse, censorship and extremism? RI CodWM DC
16
How can code reduce hospital waiting lists? ML C#1stz ML4S C#1 ML4S PO
3
Can robots unlock the secrets of the oceans? MS AS
14
How can ML be used to classify Earth images? LRN Class ML4S CICVS ML4S ACDAN
13
How can the Internet of Clean Energy work? ML4S CO ML4S I2AIoT IDL
7
Can we stop mobile phones ruling our lives? LRN Xamarin MSD FL
9
How might quantum change cyber security? ML4S Sec 7 MSR SPC
9
Business
Pitches
40 hours each
age 18
level 13
Design and build a digital product that addresses an SDG based on Software Development and Robotics and Automation.
Design and build a digital product that addresses an SDG based on Data and AI and Platforms and Cloud
Design and build a digital product that addresses an SDG based on HCI and Cyber-Security
Azure Certification

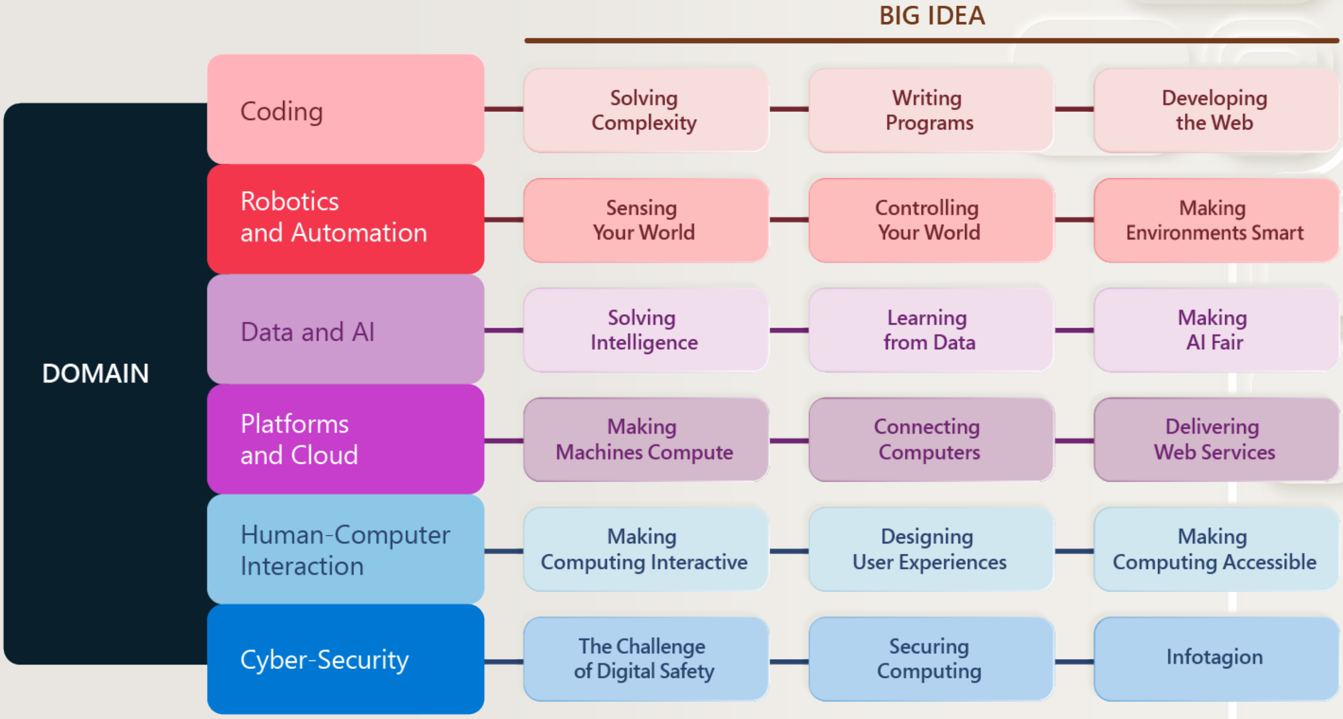
For a detailed view, please refer to the accompanying “MCSG Summary Spreadsheet”.

Chronological View

age 6 level 1	What is a computer?	How can we measure with sensors?	What is Intelligence?				Foundation
age 7 level 2				What does ' digital' mean?	How can computers and people interact?	How can you stay safe online?	
age 8 level 3	How can we make computers work for us?	How can we build a robot?	How can machines learn from data?				
age 9 level 4				How can 4bn smartphones connect?	What is a usability test?	How can you exchange secret messages?	
age 10 level 5	What Makes The Web Work?	How can we connect sensors across schools?	What rules should we give AI?				
age 11 level 6				What is in the Cloud?	How can software be more accessible?	How should you protect your online 'self'?	Product
age 12 level 7	How can I manage complexity?	How can actions be triggered with sensors?	Can machines be creative?				
age 13 level 8				How has computing evolved?	How does MR work?	How does authentication work?	
age 14 level 9	How can code help make predictions?	How can environments be controlled?	How can we make AI explainable?				
age 15 level 10	How can we use software against a pandemic?	How can admin tasks be automated?	How can bots alert farmers to weather conditions?	How can we build a LAN in a disaster zone?	How should you design interfaces for safety-critical systems?	How can data be secured?	Project
age 16 level 11	Build software to monitor air quality	Make online spaces safer for girls and women to be part of	Diagnose a medical problem with Machine Learning	Build a web service that helps/ protects an endangered species	Build a mobile app for users to identify endangered species	Build an Agriculture 4.0 solution based on IoT	
age 17 level 12	How can we build circular economies?	How can code reduce hospital waiting lists?	How can an app help provide clean water and sanitation?	How can vehicles control themselves?	Can robots unlock the secrets of the oceans?	Can cities be smart?	Business
	What is the Master Algorithm?	How can ML be used to classify Earth images?	Where has AI been fair and unfair?	How small and fast can computers sustainably	How can the Internet of Clean Energy work?	How can an Open API be used to reduce poverty?	
	Can haptics help surgeons and dentists?	Can we stop mobile phones ruling our lives?	How can computing help deal with an impairment?	Can smart contracts stop environmental damage?	How might quantum change cyber security?	How can we prevent online abuse, censorship and extremism?	
age 18 level 13	Design and build a digital product that addresses an SDG based on Software Development and Robotics and Automation.		Design and build a digital product that addresses an SDG based on Data and AI and Platforms and Cloud		Design and build a digital product that addresses an SDG based on HCI and Cyber-Security		

■ Coding ■ Robotics and Automation ■ Data and AI ■ Platforms and Cloud ■ Human-Computer Interaction ■ Cyber-Security

Conceptual View



DOMAIN	BIG IDEA		
Coding	Solving Complexity	Writing Programs	Developing the Web
Robotics and Automation	Sensing Your World	Controlling Your World	Making Environments Smart
Data and AI	Solving Intelligence	Learning from Data	Making AI Fair
Platforms and Cloud	Making Machines Compute	Connecting Computers	Delivering Web Services
Human-Computer Interaction	Making Computing Interactive	Designing User Experiences	Making Computing Accessible
Cyber-Security	The Challenge of Digital Safety	Securing Computing	Infotagion

The following sections describe the Big Ideas and the Big Questions contained within them.

Coding

Careers in Coding

It's essential that students feel motivated to develop coding skills, so before embarking on this learning journey, it's important that they understand where developing these skills can lead them.

Most modern knowledge work involves the kind of thinking that learning how to code develops:

1. **Breaking down a complex problem** or system into smaller, more manageable parts
2. **Pattern recognition –** looking for similarities among and within problems
3. **Abstraction** – focusing on the important information only, ignoring irrelevant detail
4. **Algorithms** – developing a step-by-step solution to the problem, or the rules to follow to solve the problem

Together, these steps are called 'computational thinking', and if a student thinks they can enjoy this kind of thinking, coding is something they should enjoy.

Not everyone will want to become a professional programmer, but we live in a computerised world, and understanding how to solve problems with computer code is a useful skill regardless of career choice. As people increasingly become consumers of technology at younger ages, they need to understand the principles of how things work so that they can be better informed to make decisions about complex issues surrounding

> technology. As software pervades most people's working lives, learning to code can help people interact with professional software
>
> For those who do decide to follow a career involving coding there are enormous rewards. Salaries for coding jobs can be around twice the national average wage in some countries, and demand for coding skills is, and is likely to be, very strong into the foreseeable future. In addition to this, coding is they kind of activity that suits flexible working, which again is very attractive to many people. Beyond financial and lifestyle rewards, many professional coders talk about the scale of the impact they can have as primary motivator. Coders work on some of the biggest challenges of modern times such as sustainability and climate change, healthcare and security.
>
> Explore these web pages to see how software engineers are motivated by finding opportunities to make big impacts. https://news.microsoft.com/life/why-even-one-engineer-can-have-a-big-impact/

Types of Coding Jobs

Software developer – a professional who designs software for a wide variety of purposes and clients, for use on different types of technology, such as laptops or tablets large scale (enterprise) computing systems.

Web developer – designs, revise and maintain websites using coding, working closely with other professionals such as graphic designers to create visual elements.

Computer systems engineer – specializes in designing large scale system, networks and administration solutions using complex code

Mobile app developer – a professional who often collaborates with graphic designers, writers and artists to design and implement apps for mobile devices

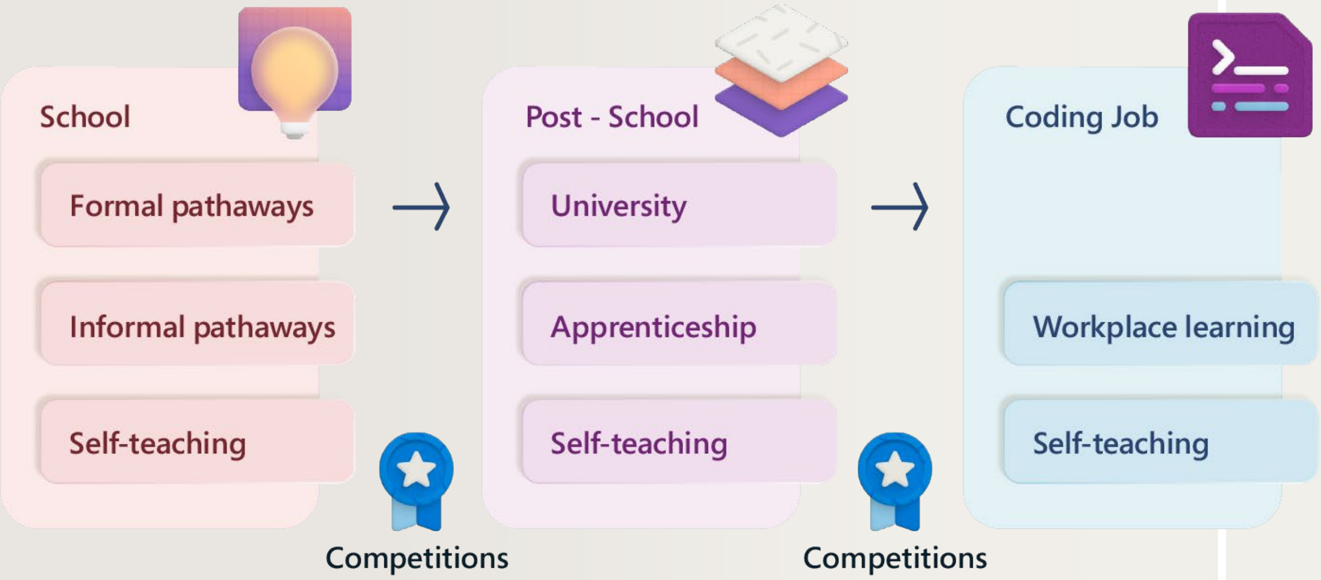
Database administrator – a professional coder with an essential role in todays' data-driven world, designing and creating databases for companies

Game developer – a coder who specializes in creating games software applications and websites collaborating with other professionals, like graphic designers, artists, voice-over artists and software engineers

Full-stack developer – a professional who codes in multiple languages to cover all aspects of large-scale computing solutions from databases to apps

How to Get a Coding Job

There are multiple pathways to a coding job. At school, there's the possibility of following a Computer Science course to gain a qualification, but also the chance to compete in hackathons and coding competitions such as Imagine Cup Junior. After school, there's the possibility of taking Computer Science at university and participating in competitions such as Imagine Cup, but for other students taking an IT related apprenticeship or going straight into work and learning on the job are also possibilities. Both formal and informal pathways offer the possibility of gaining accreditation which is often a core requirement for coding jobs. But perhaps most important of all is self-teaching – following one's interest and learning how to solve specific problems using systems such as Microsoft Learn and Stack Exchange.



School

Formal pathways

Informal pathways

Self-teaching

Post - School

University

Apprenticeship

Self-teaching

Coding Job

Workplace learning

Self-teaching

Competitions

Competitions

Explore how one young woman became a coder in this story on BBC Bitesize – https://www.bbc.co.uk/bitesize/articles/zmq3jhv

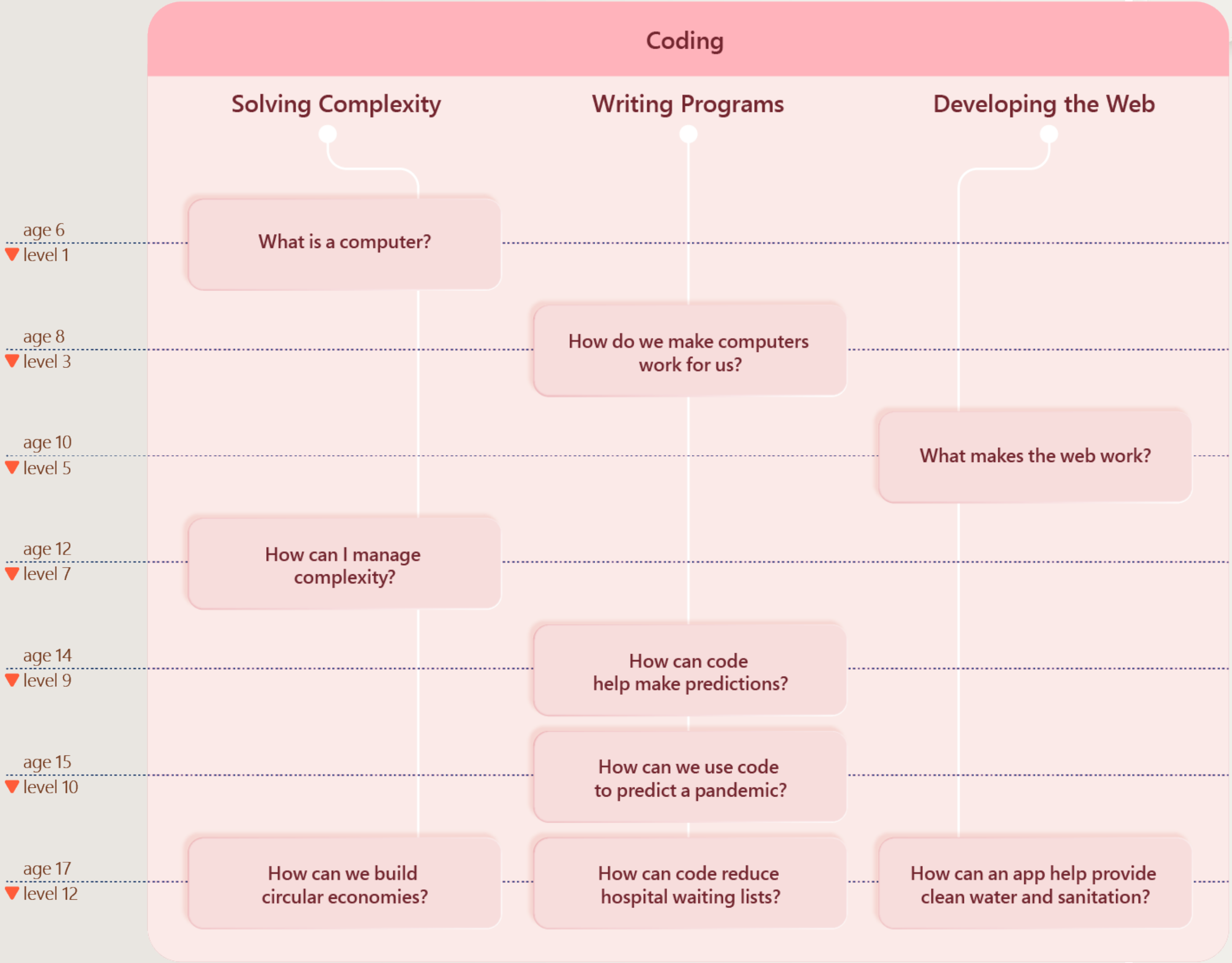
Organisation of the Coding Domain

The coding is at the heart of this CS curriculum.

In today's technology-rich world, many schools have begun teaching coding, the language we use to instruct computers. It's a skill that is in high demand, and there are many examples of schools across the world teaching it in ways that are relevant and engaging for students.

But just teaching coding is not enough. Students need to learn how to solve problems with software. This Domain uses three Big Ideas for developing and applying coding skills.

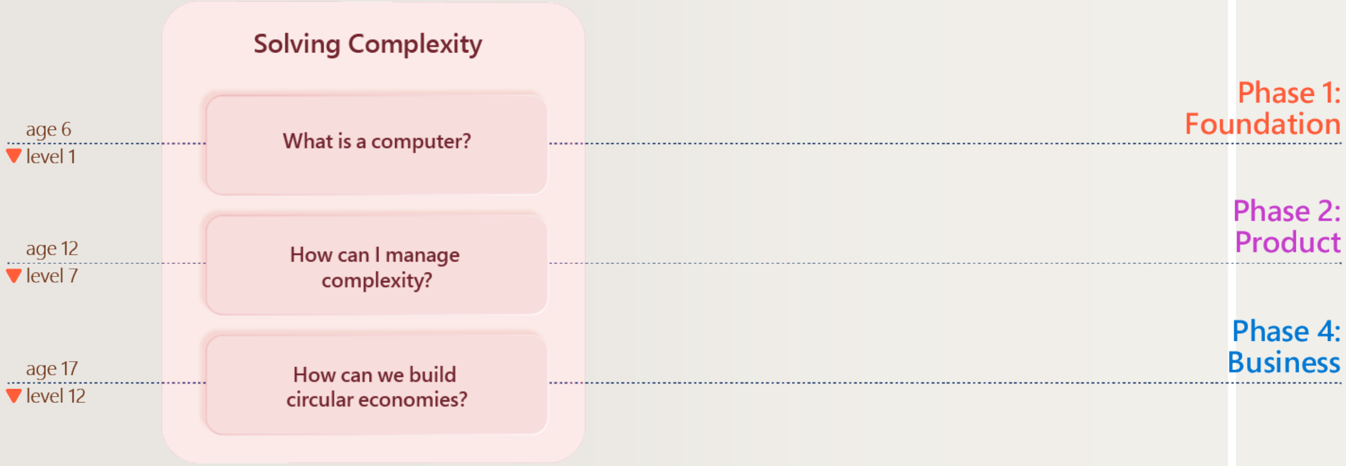
- Solving Complexity
- Writing Programs
- Developing the Web



	Coding		
	Solving Complexity	Writing Programs	Developing the Web
age 6 ▼ level 1	What is a computer?		
age 8 ▼ level 3		How do we make computers work for us?	
age 10 ▼ level 5			What makes the web work?
age 12 ▼ level 7	How can I manage complexity?		
age 14 ▼ level 9		How can code help make predictions?	
age 15 ▼ level 10		How can we use code to predict a pandemic?	
age 17 ▼ level 12	How can we build circular economies?	How can code reduce hospital waiting lists?	How can an app help provide clean water and sanitation?

The following sections examine each of the Big Ideas and the Big Questions contained within them.

Solving Complexity



The Big Idea around solving complexity includes algorithmic thinking and the use of tools and methods that support and help organise the program writing process. It is the natural conceptual and chronological starting point of the MCSG is because a prerequisite to developing software is the ability to manage the complexities involved. Furthermore, one of the main reasons that computers exist is to help humans solve complex problems.

Developing software is getting ever more complex as applications become more powerful and include increasing numbers of features. So how does a computer science professional manage complexity, deal with constant change, and implement software into larger systems in a consistent, systematic way that ensures a high-quality result? One way is to employ Process Patterns.

Process Patterns can be defined as the set of activities, actions, tasks followed in the development of software – communication, planning, modelling, development and deployment.

The foundation of Process Patterns is to understand what a ‘computer’ actually is in terms of its processes and how they can be programmed. Some of the learning activities here can be ‘unplugged’, such as those teaching students about binary numbers, searching, and sortingli.

Further down this learning path, students should learn to use tools to map knowledge. For example, this could include using Microsoft Visio for planning, to create mind maps and flow charts, or to make knowledge-graphing tools for modelling. In this learning path students will also learn about project management methodologies such as Agile and Scrum.

Key to Microsoft content

Code	Program
HSTEM	Hacking STEM, https://www.microsoft.com/en-gb/education/education-workshop/activity-library.aspxv
MDL	Microsoft Digital Literacy, https://www.microsoft.com/en-us/digital-literacy
MEE	Minecraft Education, https://education.minecraft.net/en-us/get-started
MC	MakeCode, https://www.microsoft.com
ML4S	Microsoft Learn for Students, https://docs.microsoft.com/en-us/learn/roles/student
ML	Microsoft Learn, https://docs.microsoft.com/en-us/learn
Lynda	LinkedIn Learning, https://www.linkedin.com/learning/

What is a Computer?

Domain – Coding

Big Idea – Processes and Patterns

Level – 1

Time allocation – 10 hours

Learning Goals

Main

- Understand the process of computing – input, memory, process, output –, MCSG
- Computational Thinking, MCSG – Understand fundamentals of what a computer is – hardware and software working together to process data (Map ref E4).

Supplementary

- Has a basic understanding of the word "computer"
- Knows what an algorithm is
- Can explain the difference between hardware and software
- Knows what binary is and why its relevant to computing
- Can build a simple circuit and switch

Concepts

- A computer can carry out 'thinking' type activities based on instructions called algorithms and binary data
- Software requires hardware to work in the same way as a pen requires a paper for writing.
- A computer operates by using electronic components, data and algorithms
- The core operation of computing involves millions of very small electrical switches
- Digital data can be saved onto a computer's memory: voices, images, video, words and numbers

Challenges

Computational Thinking – Algorithmic thinking e.g. shopping list, story sequencing, dance moves, party planning

Data Literacy – Binary numbers, image representation

Design Thinking – Connect switches together and explore different combinations

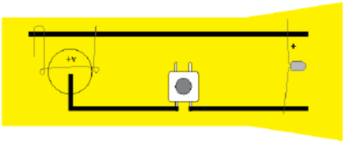
Example Lesson Outlines

1. *What is a Switch?*

Get the students to make a flashlight.

After succeeding in the main task, students can develop their Design Thinking by connecting switches together to explore different on-off combinations

Download instructions here: **H-STEM FL** https://aka.ms/Flashlight



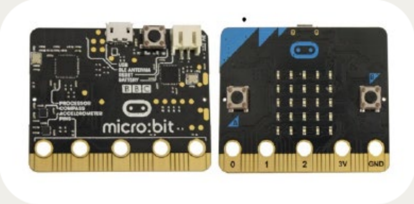
2. *How are Switches Linked to Software?*

Connect the idea of a flashlight being on and off to binary – i.e., off = 0, on = 1. Get the students to work out numbers from flashes – e.g.:

8	4	2	1	
0	0	1	0	2
0	1	1	0	6

3. *What is an Electronic Circuit?*

Using the BBC micro:bit illustrate and explain how circuits are fundamental to computing. https://microbit.org/get-started/user-guide/overview/#new-micro:bit-with-sound



Explore is meant by ‘micro’ and ‘bit’ in the name micro:bit?

- “Micro” means microprocessor – a circuit made up of millions of tiny switches.
- “Bit” means binary digit – a 0 or a 1.

Put them together and you have a computer.

In the same way the terms micro and soft in the name ‘Microsoft’ refer to microprocessors and software.

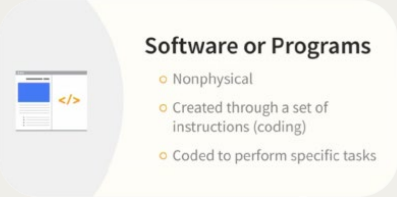
4. *What Can You do With a micro:bit?*

Explore some of the basic functions of the micro:bit starting with the “Out of box” experience.

5. *What is Hardware and Software?*

Get the students to follow the video below and then demonstrate different pieces of hardware, and software.

MDL WWC 1 https://www.linkedin.com/learning/working-with-computers-and-devices/what-is-a-computer?autoplay=true&resume=false



Software or Programs

- Nonphysical
- Created through a set of instructions (coding)
- Coded to perform specific tasks

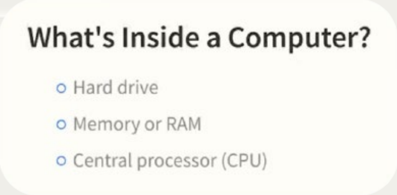
Computational Thinking – to help explain what software is, get the students to create algorithms, e.g., shopping list, story sequencing, dance moves, creating patterns.

Write the algorithm in the Minecraft Instruction Book, https://education.minecraft.net/en-us/challenges/minecraft-instruction-book

6. *What Hardware Parts Make up a Computer?*

Explore and demonstrate different parts of the computer and get students to be able to confidently name different parts. Use the video below as a guide.

https://www.linkedin.com/learning/working-with-computers-and-devices/what-parts-make-up-a-computer?autoSkip=true&autoplay=true&resume=false



What's Inside a Computer?

- Hard drive
- Memory or RAM
- Central processor (CPU)

7. What Forms Can Computers Come in?

Students will have already worked with one kind of computer – the micro:bit, so they should now be ready to explore other types of computers. Use the video below as a guide.

https://www.linkedin.com/learning/working-with-computers-and-devices/understand-types-of-computers?autoSkip=true&autoplay=true&resume=false

8. What is Computer Storage?

Data Literacy – The files, photos, programs, music, and movies that we use has to be stored somewhere. Get students to explore all the main kinds of digital data and how they are stored, and get them to move data from one type of media to another.

Use the video below as a guide.

https://www.linkedin.com/learning/working-with-computers-and-devices/add-external-storage-to-a-computer?autoSkip=true&autoplay=true&resume=false

9. How Can Computers Communicate With One Another?

How can data be transferred between computers or across parts of computers? Get students to pair micro:bits, PCs and peripherals using Bluetooth.

Use the video below as a guide to using Bluetooth.

https://www.linkedin.com/learning/working-with-computers-and-devices/add-external-storage-to-a-computer?autoSkip=true&autoplay=true&resume=false

10. Skills and Understanding Check

Set up practical tasks that enable students to both work together and individually so solve problems and answer questions related to the content covered in lessons 1-9 here.

Supporting Content

- **H-STEM MPLU**, Making party lights to understand electrical circuits and switches, Excel. Micro:Bit, https://docs.microsoft.com/en-us/learn/educator-center/instructor-materials/make-party-lights-understand-electrical-circuits-switches
- Computer Science Unplugged (**CSU** 1), Binary numbers – https://www.csunplugged.org/en/topics/binary-numbers/
- Computer Science Unplugged (**CSU** 11), Class simulation of a computer – https://classic.csunplugged.org/class-simulation-computer-unfinished/
- Code.org (**CSD** Unit 1) – Problem Solving. What is a computer? IO, processes and storage: https://studio.code.org/s/csd1-2019
- **BBC** Bitesize Introduction to computational thinking – https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/2
  https://learn.microsoft.com/en-us/training/educator-center/instructor-materials/hacking-stem?source=recommendations
  https://www.khanacademy.org/computing/ap-computer-science-principles/computers-101/introducing-computers/a/what-is-a-computer
  https://kids.britannica.com/kids/article/computer/352990
  https://kids.kiddle.co/Computer
  https://cacm.acm.org/blogs/blog-cacm/231993-what-children-want-to-know-about-computers/fulltext

How Can I Manage Complexity?

Domain – Coding

Idea – Processes and Patterns

Level – 7

Time allocation – 10 hours

Learning Goals

Main

Computational Thinking, MCSG, manage complexity using established methods, procedures and tools for structuring and managing projects; and producing, versioning and sharing code (Map ref E10).

Supplementary

- Gain confidence in Computer Science by recognising that all computing problems can be addressed through decomposition and abstraction, and established methods
- Use the principles for writing, testing, and sharing code
- Create structured processes and plans including using visualization methods

Concepts

- Complexity can be managed through decomposition and abstraction, and standard methods
- Computer systems are organised using established principles, and developed using established methods
- Visualising systems is an important way to manage complexity
- A clear understanding of the problem is essential throughout the development process
- Different tools and methods need to be applied with precision to solve specific problems

Challenges

Computational Thinking – Demonstrate an understanding of established processes used to manage and reduce complexity

Data Literacy – Show an understanding of how data enters and is processed in a computing system

Design Thinking – Draw flow chart diagrams

Example Lesson Outlines

Computing systems can seem overwhelmingly complex, but asking the right questions and choosing the right methodology and approaches will enable students to manage complexity and produce solutions to computing problems.

1. What Problem Are We Trying to Solve?

Complexity in Computer Science can be addressed in two simple stages –

1. Understand and categorise the problem
2. Select and use appropriate methods and tools to solve the problem

Let’s begin with understanding and categorising the problem.

There are few more important questions in the world of computing than ‘what problem are we trying to solve?”

Students need to be able to ask this question confidently and frequently as they manage computing complexity so present students with a range of business and sustainability problems and ask –

- Is this problem solvable through computing?
- What is the nature and parameters of the problem?
- What would successful solution to the problem look like?

Get students to clearly define different problems

Supporting content – Microsoft Learn, 'Plan your project like a pro', **ML PYP** https://docs.microsoft.com/en-us/training/modules/plan-app-project/

2. How Do We Categorise Computing Problems?

Once students have learned the importance of asking "what problem are we trying solve", the next step is for them to learn how to put the problem in the right domain. Present the students with different types of problems and ask them to classify these in terms of the following MCSG domains –

- Data & AI
- Platforms & Cloud
- Human-Computer Interaction
- Robotics & Automation
- Cyber-Security

3. How Do We Establish Solution Project Goals?

When students have learned to frame different types of problems, the next step is to give them the tools to organise a solution project starting with establishing clear goal. Key topics should include:

Delivering Quality

- The 'Iron Triangle' of project management – Scope, Cost, Time, Quality

Minimising Risk

- "Scope creep"
- New technology
- Security

Maximising Benefits

- End-user benefits
- Efficiency and performance
- Business benefits – the business case

4. How Are Computing Solution Projects Structured?

With an understanding of how to ensure that a computing project has clear goals, students can progress to learning how projects are structured and managed. Key topics should include:

Project Planning

- Scheduling and milestones
- Resources, budgets, and constraints
- Risk management

Executing and Monitoring a Project

- "The stack"

- The Waterfall Software Development Life Cycle Model
- Dev Ops Pipeline – plan > code > integrate > test > release > deploy > operate > feedback
- Visualisation methods
    - Pipelines
    - Block diagrams
    - Standard symbols
    - Route Cause Analysis

Post-project Review

- Measuring success
- Improvements for next version
- Lessons learned

People often process and retain information much more efficiently when it is presented in the form of an image instead of text so a key focus here should be on developing visualisation skills, including using a variety of techniques to illustrate what a problem entails, and how to approach its solution. Visualisation skills combine abstraction with decomposition.

Visio is an important tool to master, so students should watch the following tutorial – **BTV** https://support.microsoft.com/en-us/office/beginner-tutorial-for-visio-bc1605de-d9f3-4c3a-970c-19876386047c?ui=en-us&rs=en-us&ad=us

5. How Can We Plan Software Development?

Most computing developments will involve software development – coding – at some level. Before any code is written it's essential to be able to produce flowcharts to decompose the problem, clarify the functions of the software being written, and show how the software itself will work.

For example, measuring rainfall across a region

Query the database in zone A to get rainfall in mm for past 30 days

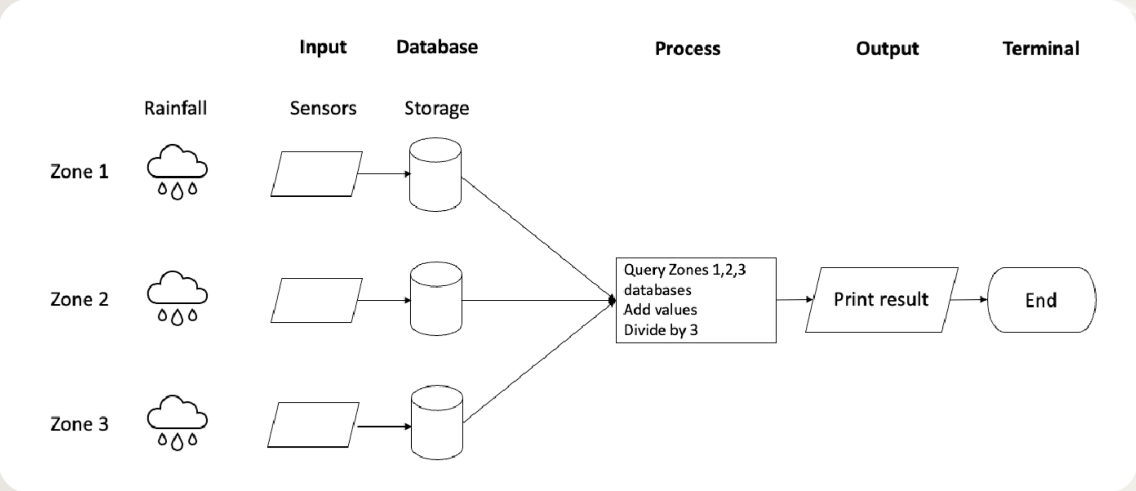
Query the database in zone B to get rainfall in mm for past 30 days

Query the database in zone C to get rainfall in mm for past 30 days

Add each number together and divide by 3

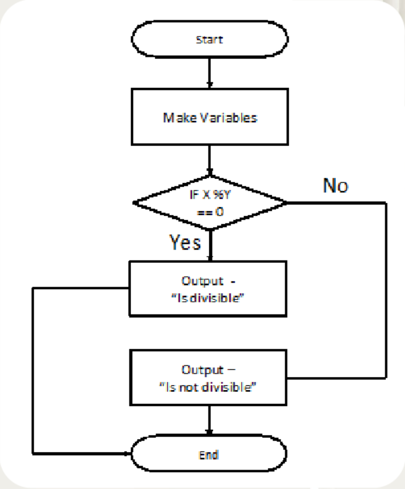
Display the result

This algorithm can be represented diagrammatically like this –

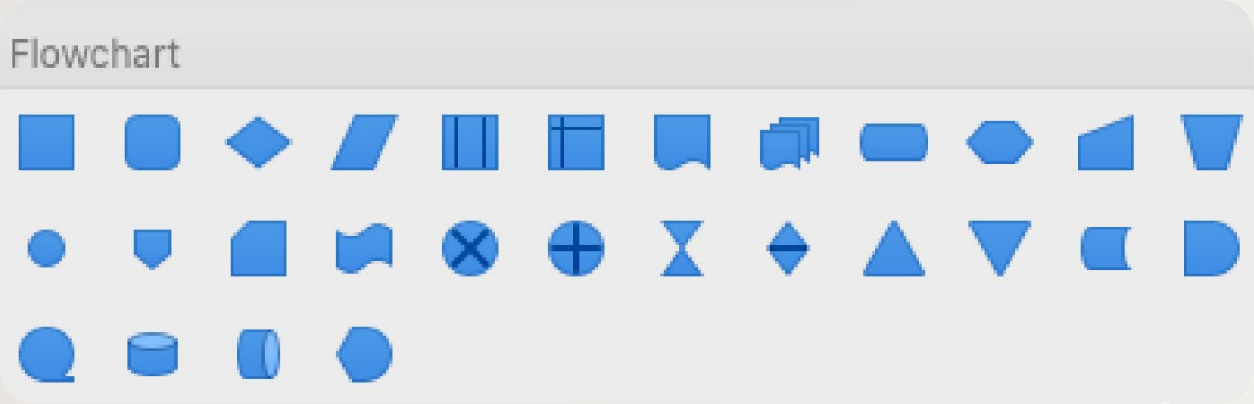


For a piece or pieces of code, a flowchart can identify the essential steps and show the bigger picture of the process. It organises the tasks in chronological order and identify them by type, e.g. input/output, process, decision, data etc.

For example:



Supporting content and tools – Microsoft PowerPoint has a library of standard flow-chart icons in its Shapes library –



Learn more about the international set of flowchart standards called ISO 9001 here – **LI FC** https://www.linkedin.com/pulse/how-create-iso-90012015-process-flowchart-ayman-aburawwa/?trk=pulse-article_more-articles_related-content-card

6. Choosing Languages

Managing complexity is about understanding and categorising the problem then selecting and using appropriate methods and tools to solve the problem.

But here are hundreds of different languages so students should know how to choose them to solve different types of problems.

The top 4 languages used worldwide[lii] are Python, C, C++ and Java, and students should become familiar with these. We should also consider a language to work with databases, such as SQL.

Key topics here should include:

Compiled vs Interpreted. Theoretically, languages *per se* are neither compiled nor interpreted, but in practical implementation languages are often classified as one or the other. Code in a compiled language is converted directly into machine code that will work with specific hardware. Code in an interpreted language is converted in an intermediary application that runs on any hardware. Generally compiled languages are faster than interpreted.

Object Orientated Programming (OOP) is a module-based approach to writing software which allows a range of entities to defined as 'objects' and packaged as 'classes'[liii]. Benefits include reusable code, easier troubleshooting and maintenance, and flexibility.

Python is an OOP interpreted language, with a lot of libraries and modules which means that many complex tasks can be executed very simply using these libraries and built-in functions. Python is extensively used in Data Science and Machine Learning.

C is not an OOP but allows a complex program to be broken into simpler programs called functions. It is commonly used on the largest supercomputers to the smallest microcontrollers, and in games, and applications requiring calculations. Because C uses fewer instructions, it runs faster than other programming languages.

C++ is derived from C and shares common syntax. It is compiled, OOP, fast and like C, gives developers direct access to hardware. It is used to build operating systems, GUI-based applications such as Photoshop, browsers, and cloud systems. It is platform dependent – in other words, C++ apps are written for use on a single platform and must be rewritten for use on other platforms.

Java – Interpreted, platform-independent "write once, run anywhere". It's used a lot for mobile app development – for example, most Android applications are built using Java. Many popular games are written using Java[liv] including the first versions of Minecraft. Java is also used for web apps.

SQL – Structured Query Language – is a programming language focussed on the management of data. It is used for finding, retrieving, and manipulating data stored in even the largest databases, and for classifying, estimating, and forecasting.

Students should be able to select the right language for a given a range of scenarios and should be presented with a range of real-world scenarios and asked to select the right kind of language for the task. They can also be asked to produce a table comparing the attributes of each language.

Another task that students could be asked to do here is write and compare code to print "Hello World" in each of the languages above.

Visual Studio Code provides a coding environment for all the languages mentioned here – download it at https://code.visualstudio.com

7. What Are The Principles Behind Writing Code?

Software can sometimes contain millions of lines of code and appear overwhelming complex. However, even the most complex of software is written using accepted principles. At this stage, students' ability to understand and explain a program is much more important than the ability to produce working but incomprehensible code, and understanding this is based on knowing the following principles:

- **Sequencing** – doing one step after another
- **Branching** – doing either one thing or another depending on whether something is true or false, 'if-then-else', using Boolean expressions
- **Nesting** – the idea that code that performs a particular function can be contained within code that performs a broader function
- **Repetition** – iterative loops or recursion, e.g. a 'for' or 'while' statement
- **Interaction** with input/output, event-based programming – i.e. human user or sensor inputs
- **Debugging** – finding and correcting errors in code
- **Evaluation** – reflecting thoughtfully on their program, including assessing its correctness and fitness for purpose; understanding its efficiency; and describing the system to others
- **Abstraction** – wrapping up a computation in a named abstraction, so that it can be re-used. The most common form of abstraction is the notion of an "function", "object", "class", "procedure", or "library"

- **Objects** – a coding entity that possess properties or attributes and are capable of some functionality or behaviour

Students should be asked to produce some code to show that they understand these concepts, for example, asking them to write code to draw a square. This should call a procedure to draw a line; the procedure to draw a line calls a procedure to paint a pixel; the procedure to paint a pixel calls a procedure to calculate a memory address from an (x,y) pixel coordinate. In this activity, a student should recognise if they are writing more or less the same code repeatedly and abstract this for example with 'draw a square of size N'.

Object Orientated Programming

Students should also be asked to produce code that shows they understand Object Orientated Programming (OOP) principles. OOP can be a hard concept to grasp, so it's worth looking at in more detail here.

In coding, the basic building block in object-oriented design is an object. Objects possess properties or attributes and are capable of some functionality or behaviour. For example, in the Angry Birds game, the birds do not have any feet or feathers to flutter – they mostly fly as projectiles, bounce around when hit, and so on. Each domain in the Angry Birds program has a set of objects that typically interact with one another. Birds in the game, for example, have properties like their colour, size, position, velocity, type of bird, and so on. The game's green pigs are green, sometimes wear helmets, and mostly sit in one place. The "castles" or structures themselves are objects.

To demonstrate an understanding of OOP, students can create several objects from a class – say the class "Ball", for example. Here, balls can be labelled – if using three balls, they could be labelled b1, b2, and b3. Each of these balls has the same internal structure as defined by the class and yet they each are distinct objects distinguished by the different values of their properties. Ball b1 might be small in size, green in color as opposed to b2, which might be large and blue.

For example, in Java, to define Classes we follow this syntax:

```
class ClassName {

// Attributes

…

// Behaviors

…

}
```

Here's how the Ball class can be defined:

```
class ball { // Define the ball class

// Attributes

float x, y; // the x and y location

color ballColor; // its color

float radius; // its radius in pixels

} // class ball
```

This code defines a class called "ball". Attributes of the class are defined as variables. Thus, "x" and "y" specify the x and y coordinates of the ball's position, ballColor, and radius will be used to store the color and size, respectively. To demonstrate an understanding of OOP, students could go on to develop this code into a working demonstration.

Supporting content: Computer Science Principles course, Unit 1 – Introduction to AP Computer Science Principles, MC CSP AP 1 https://microsoft.github.io/makecode-csp/unit-1/introduction

8. How Can Code Be Made to Work?

Code must be made to work, and for all but the simplest programs students need to learn that writing is almost always followed by debugging. The key to coding success is for students to develop a systematic approach to detecting, diagnosing and correcting faults, and to develop debugging skills, including reading, and understanding documentation and explaining how code might or might not work.

Students should be given broken code to fix that gives them the following learning opportunities:

1. Isolating or localising faults by 'commenting out' lines of code that may contain errors then re-running the code
2. Adding comments to make code more human readable
3. Adding error checking code to check internal consistency and logic
4. Choosing test cases and constructing tests
5. Explaining how code might or might not work

9. How Can Code Be Shared and Deployed?

A key part of code development is sharing because this allows other people to review, input their own specific expertise, and make improvements to the code. To do this successfully requires students to learn two skills.

1. Versioning
   Each version of the code and its dependent components needs to be labelled clearly. Common practice is to append the name of the software with three numbers. E.g. if the version of the software is "2.7.9" the '2' means major changes that breaks the previous version, the '7' means the 7th set of minor changes that don't break the current version, and the '9' means this version has had 9 patches or bug fixes.

2. Git & GitHub
   Git is a software version control system that helps developers keep track of code changes. GitHub is a platform used to share code with others. The basic workflow of GitHub is:

   Step 1 – Users modify a file from the working directory

   Step 2 – Add these files to a 'staging' area

   Step 3 – Perform a 'commit' operation that moves the files from the staging area, and 'push' the files, to store the changes permanently

Supporting content – Microsoft Learn, Manage the lifecycle of your projects on GitHub **ML GHL**
https://docs.microsoft.com/en-us/training/paths/manage-project-lifecycle-github/ (introduction to GitHub)

Deployment

Once code has been written and debugged, it next needs to be shared and implemented. Code must be deployed in a way that doesn't affect already functioning features and can be available for end-users, so implementation often starts on a 'staging server or environment' which models the 'production' (end-user facing) environment. After testing, the code can run in the production environment.

Students should be given the opportunity to upload code to GitHub, version it, allow others to modify, and perform modifications to other student's code in GitHub.

10. Skills and Understanding Check

Set up practical tasks that enable students to both work together and individually to solve problems and answer questions related to the content covered in lessons 1-9 here.

Supporting Content

- Go deeper into coding – Computer Science Principles course, 8 units, https://makecode.com/csp
- Visio Essential Training, https://www.lynda.com/Visio-tutorials/Visio-2019-Essential-Training/743151-2.html
- Beginner tutorial for Visio, https://support.microsoft.com/en-us/office/beginner-tutorial-for-visio-bc1605de-d9f3-4c3a-970c-19876386047c?ui=en-us&rs=en-us&ad=us
- Project Management for Middle School, https://www.edutopia.org/article/project-management-middle-school

How Can We Build Circular Economies?

Domain – Coding

Big Idea – Solving Complexity

Level – 12

SDG – 12. Responsible consumption and production

Time allocation – 6:40 hours

Learning Goals

Main

Design Thinking, MCSG, Apply project management and sustainability principles to apply software to a sustainability problem (Map ref E22).

Supplementary

- Start to learn about OOP and DRY
- Apply Business Model Canvas, MCSG
- Collaboratively deploy and manage software artefacts on a code management platform, MCSG
- Apply Business Model Canvas, MCSG

Concepts

- A Circular Economy is an economic system aimed to design-out waste and pollution and regenerate natural systems
- A Business Process Map (BPM) is a flow-chart representing the key processes, entities, and decision points of a business process

Challenges

Computational Thinking – Collaboratively develop software that can turn an unsustainable organisation into a Circular Economy.

Data Literacy – Analyse the cumulative energy and waste outputs of a process through time and measure improvements

Design Thinking – Represent the 'as-is' lifecycle of a manufactured good in a business process map and re-envision it as a circular process that minimizes waste

Example Project Outline

One area of high complexity that Computer Scientists need to master is sustainability, particularly with respect to climate change. However, through applying established scientific and mathematical principles scientists have been able to reliably model the Earth's ecosystems and conclude that human activity is currently unsustainable. This project is about converting the complexity of organisational processes into practical actions that lead to sustainability.

Brief

Develop a 'sustainability dashboard' and design the underlying software solution that can be used to turn your school into a Circular Economy.

This brief does not require a fully functioning solution, but does require the following:

- A coded calculator that outputs an overall improvement score
- Mock-up of a dashboard showing overall improvement score and measures of contributing factors
- Clearly communicated system architecture for a full-scale solution

Ideally, OOP will be used in the coded calculator.

Project Framework

Phase 1, Understand OOP – 1 hour

Students should start by acquiring an understanding of what OOP is and how they might be able to use it in answering the brief.

They can get an introduction to object-oriented programming with Python here: **ML OOP** https://docs.microsoft.com/en-us/training/modules/python-object-oriented-programming/

Phase 2, Ideation – 1 hour

The first part of the project is to follow steps 1-5 set out in "How Can I Manage Complexity?" in the Product phase of the MCSG –

- Define what problem we are trying to solve?
    - What does the research say about Circular Economies, and what are the implications for the operations of a school?
    - What kind of functions would we want a dashboard to have in this context?
    - What kind of software solution is required for dashboards to work?
- What categories of computing will we need to focus on?
    - Will the solution involve using sensors and/or automation?
    - Where does data, data visualisation and predication fit in?
    - What type of platform should be used?
- What are the goals of the solution?
    - What should change in your school as a result of the use of the solution? For example:
        - Lower environmental footprint
        - Reduced costs
        - A healthier school community
    - How would success be measured?
    - What needs to be counted to calculate the overall success measure?
- How should the project be structure?
    - In the short time available, how should the team and the project be organised?

Students will need to look holistically at the operation of their school, and some suggested categories for their Circular Economy dashboard are below.

Process		
CO2 emissions	**Sustainable travel**	**Sustainable food**
Carbon literacy	Bus travel	Plant-based food consumption
Carbon management plan	Car sharing and electric car pooling	School-produced food consumption
Infrastructure	Electric car use	Locally sourced food consumption
Green roof and walls	Bike usage	Reduction of food waste
Trees planted	Walking	Reusable mugs and packaging
Energy efficiency	Waste Reduction	Food waste segregation
Energy saving measures	Shared equipment inventory system	Composting
Energy-efficient infrastructure	Reduction of use of consumables	
Water consumption	Furniture for reuse store	
Water saving measures	IT reuse centre	
	Recycling	

Students should research Microsoft Cloud for Sustainability, MS CFS https://youtu.be/2alGZWbeaNI and note that this is for emissions only, and their brief here is for the full Circular Economy of their school, not just CO2 emissions.

This research from Manchester University sets out a methodological framework for the implementation of Circular Economy thinking for universities, but the document contains some useful information that can be applied to Circular Economies in schools. The students can be asked to 'speed read' this document to extract the key points and draw implications for the brief that they have been set – https://www.research.manchester.ac.uk/portal/files/93236310/CE_thinking_in_HEIs.pdf

Phase 3, Build– 3 hours

The first part of the build process is to follow steps 7-9 set out in "How Can I Manage Complexity?" in the Product phase of the MCSG –

- How should the code be written?
    - What types of calculations need to take place?
    - What is the optimal platform?
    - What coding principles need to be applied to the calculation component?
- How can the code be de-bugged?
- How can code be shared and deployed?
    - How can colleagues and others contribute to the code development?
    - How would the solution be deployed if taken to market?

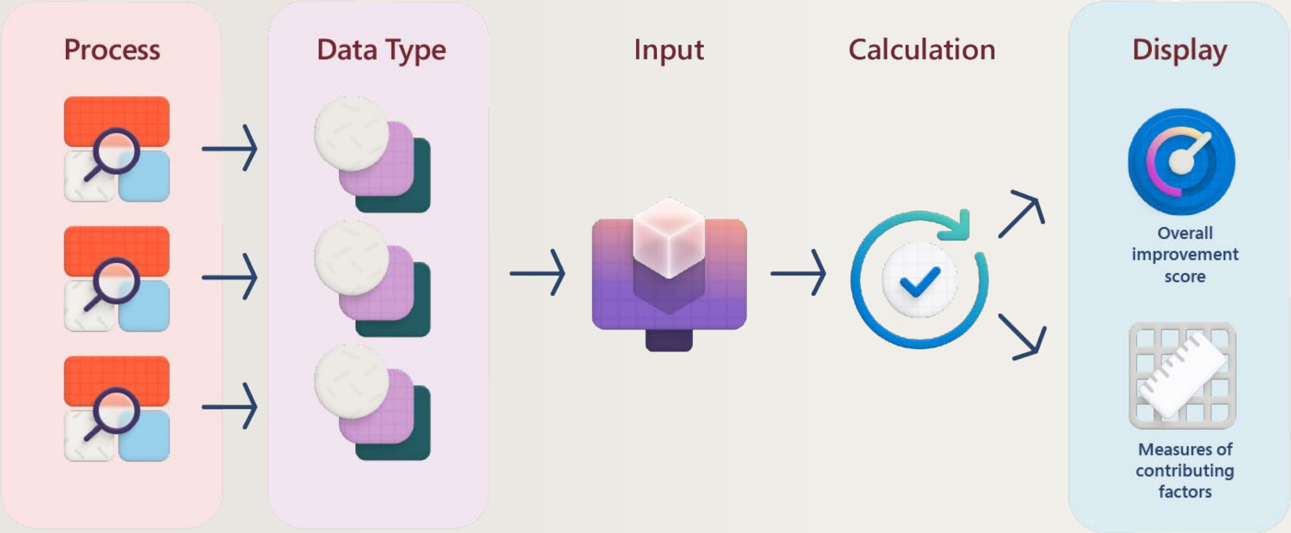
In this phase there are several problems that need to be solved as suggested in the diagram below.

First, processes need to be selected, and here, for example, students could use the 80:20 principle – how can we get 80% of the data we need from 20% of the processes that are available for us to analyse?

Next students need to analyse the type of data generated by each process in terms and understand how it's going to be collected – for example, will the data be collected by sensors, or inputted by people, or does it already exist digitally in within the school's IT system?

The core of the solution is a calculator – an algorithm that aggregates, weighs and coverts data into i) an overall Circular Economy improvement score, and ii) allows users to see what contributing factors may be affecting the score.

Finally, the results of these calculations need to be outputted in a dashboard that can be read and understood by decision makers and stakeholders.



Phase 4, Present and Evaluate – 1 hour 40 mins

Assessment of this project is via group evaluation of projects, and teacher assessment of contributions to the team by individual members.

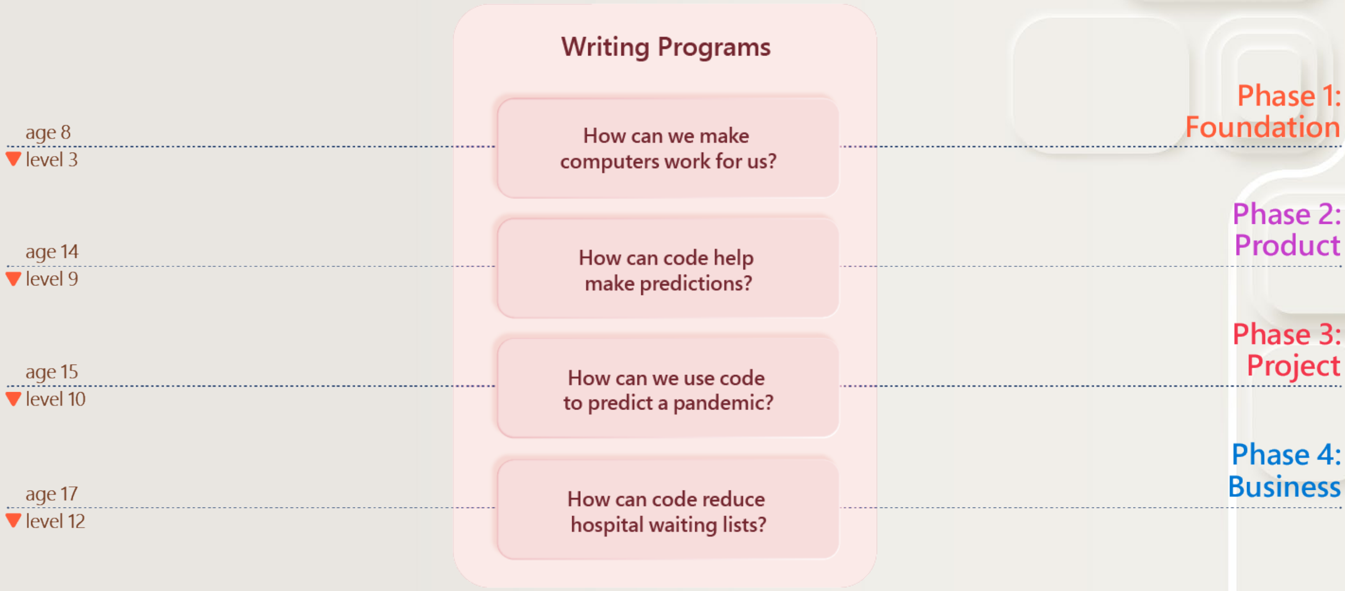
Each team, ideally 5 diverse members of mixed ability, is given 5 mins to present their solution in slides and a demo, followed by 5 min Q&A. All other students in the class are encouraged to ask questions and one way to do this is to award team points for good questions.

Ideally, each team will make a video of their solution and presentation for future reference.

Supporting Content

- **ML4S I2G**, Introduction to Git, https://docs.microsoft.com/en-us/learn/modules/intro-to-git/
- **GSDP**, GitHub Student Developer Pack, https://education.github.com/pack
- Lynda Ag, Agile Fondations, https://www.lynda.com/Project-Management-tutorials/Agile-Foundations/761929-html?srchtrk=index%3a4%0alinktypeid%3a2%0aq%3aagile%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2

Writing Programs


Writing Programs
age 8
level 3
How can we make computers work for us?
Phase 1: Foundation
age 14
level 9
How can code help make predictions?
Phase 2: Product
age 15
level 10
How can we use code to predict a pandemic?
Phase 3: Project
age 17
level 12
How can code reduce hospital waiting lists?
Phase 4: Business

This Big Idea focusses on writing software, progressing from block-based visual, to scripted, then to object-oriented programming and languages.

Building on their understanding of how a computer works acquired in “Solving Complexity”, the foundation of “Writing Programs” is understanding how humans can get computers to work for us through the basics fundamentals of producing code.

As they progress through this Big Idea, students will cover algorithmic thinking and language concepts including syntax, assignment, control structures, variables, data types, data structures, modelling, functions, loops, statements, operators, and conditional execution.

As students progress further and their skills develop, they will be set increasingly open-ended authentic problems to solve through writing code.

How Can We Make Computers Work For Us?

Domain – Coding

Big Idea – Writing Programs

Level – 3

Time allocation – 10 hours

Learning Goals

Main

- Write sequences, events, loops, and conditionals, *MCSG*
- Create programs that include sequences, events, loops, and conditionals, *CSTA 1A-AP-10*
- Use post-tested loops e.g. ‘until’, and a sequence of selection statements in programs including an 'if', 'then' and 'else' statement, *CAS P&D 3*

Supplementary

- Understand that humans instruct computers what to do through programming

- Understand that code is determined by design goals
- Assemble instructions as building blocks into a program

Concepts

- Essential functions of a computer: input/memory/processor/output
- Structure of a computer program with input, conditions, and output
- Hardware vs. Software

Challenges

Computational Thinking

- Use code to create a moving system
- Use multiple moving parts to create an animated feature
- Code Variables, Sprites, Motion, Loops

Data Literacy

- Count the arrangements, possibilities, and configurations (combinatorics)
- Find the largest, smallest, best (optimal) solutions

Design Thinking – Design a landscape – such as a maze – that your friends can 'walk' through

Example Lesson Outlines

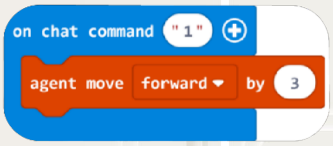
1. *Coding Fundamentals 1*

Block 1, Lesson 1 – Animal Research Center (ARC): Basic Moves and the Agent

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Learn basic navigation in Minecraft Education
- Become familiar with Code Builder and MakeCode blocks

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/5320521929977595-Coding%20Fundamentals_Block%201%20Lesson%201_Basic%20Moves%20and%20the%20Agent.pdf



```
on chat command "1" ⊕
    agent move forward ▾ by 3
```

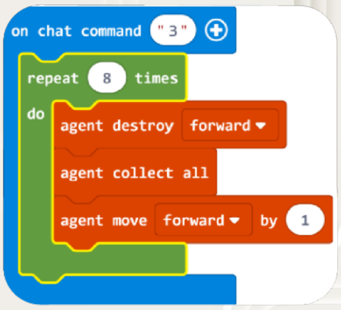
2. *Coding Fundamentals 2*

Block 1, Lesson 2 – Sea Turtle Assistance: Algorithms, Sequencing, & Loops

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Use pseudocode to help create algorithms
- Test and debug code
- Use coding concepts of sequencing, pattern recognition, and loops

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/1952294412733588-Coding%20Fundamentals_Block%201%20Lesson%202_Sea%20Turtle%20Assistance.pdf



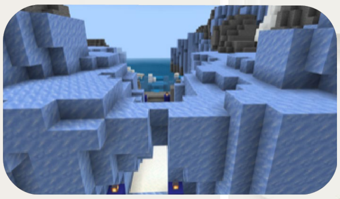
```
on chat command "3"
  repeat 8 times
  do
    agent destroy forward
    agent collect all
    agent move forward by 1
```

3. *Coding Fundamentals 3*

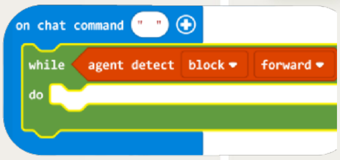
Block 1, Lesson 3 – Polar Bear Rescue: Sequencing and Conditional loops

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Use the coding concepts of loops and conditionals to complete tasks



Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/1972279697524382-Coding%20Fundamentals_Block%201%20Lesson%203_Polar%20Bear%20Rescue.pdf



```
on chat command " "
  while agent detect block forward
  do
```

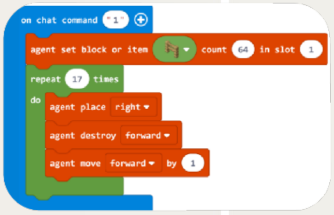
4. *Coding Fundamentals 4*

Block 1, Lesson 4 – Panda's Protected Habitat: Loops and Nested Loops

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Use the coding concepts of loops and nested loops to complete tasks

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/0257429388368151-Coding%20Fundamentals_Block%201%20Lesson%204_Pandas%20Protected%20Habitat.pdf



```
on chat command "1"
  agent set block or item count 64 in slot 1
  repeat 17 times
  do
    agent place right
    agent destroy forward
    agent move forward by 1
```

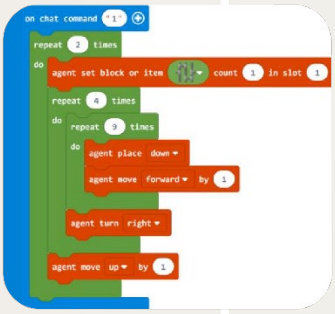
5. *Coding Fundamentals 5*

Block 1, Lesson 5 – Gray Wolves: Loops and Conditional Loops

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Use the coding concepts of loops and nested loops to complete tasks

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/6833267756641344-Coding%20Fundamentals_Block%201%20Lesson%205_Gray%20Wolves.pdf



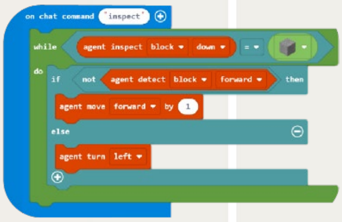
```
on chat command "1"
  repeat 2 times
  do  agent set block or item ▾ count 1 in slot 1
      repeat 4 times
      do  repeat 9 times
          do  agent place down ▾
              agent move forward ▾ by 1
          agent turn right ▾
      agent move up ▾ by 1
```

6. *Coding Fundamentals 6*

Block 2, Lesson 1 – Minecraft Space Center

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Reinforce the use of the coding concept of repeat loops
- Learn the function and use of new MakeCode blocks for the Agent
- Utilize the new coding concepts of while loops
- Utilize the new coding concept of conditionals

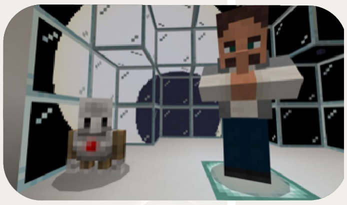


```
on chat command "inspect"
  while  agent inspect block ▾ down ▾  = ▾  ▾
  do  if  not  agent detect block ▾ forward ▾  then
        agent move forward ▾ by 1
      else
        agent turn left ▾
```

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/3561179887482767-Coding%20Fundamentals_Block%202%20Lesson%201_Minecraft%20Space%20Center_REVISED.pdf

7. *Coding Fundamentals 7*

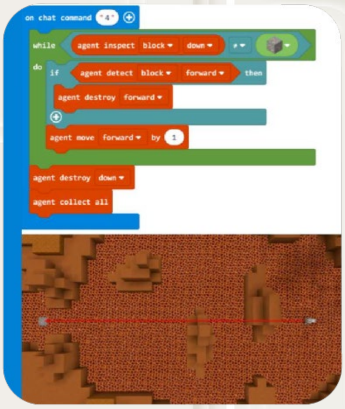
Block 2, Lesson 2 – Mercury and Venus: Reinforcing Loops and Conditionals



MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Follow coding instructions to solve puzzles without usual visual Minecraft support
- Use operators, conditionals, and while loops in their programs

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/7930799685461426-Coding%20Fundamentals_Block%202%20Lesson%202_Mercury%20and%20Venus.pdf



```
on chat command "4"
  while  agent inspect block ▾ down ▾  ≠ ▾  ▾
  do  if  agent detect block ▾ forward ▾  then
        agent destroy forward ▾
      agent move forward ▾ by 1
  agent destroy down ▾
  agent collect all
```

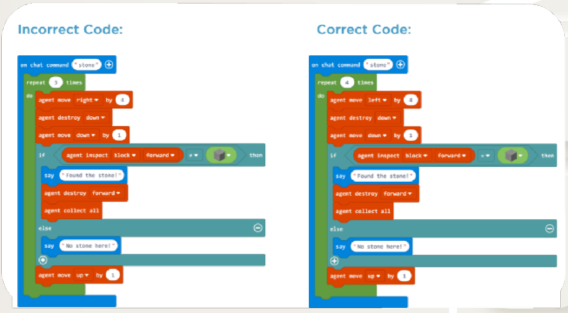
8. *Coding Fundamentals 8*

Block 2, Lesson 3 – Earth's Moon and Mars: Debugging

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

- Use an effective debugging process
- Exercise persistence and attention to detail while solving coding puzzles

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/6417758637958634-Coding%20Fundamentals_Block%202%20Lesson%203_Earths%20Moon%20and%20Mars.pdf



Incorrect Code:

```
on chat command "stone"
  repeat 3 times
  do
    agent move right by 4
    agent destroy down
    agent move down by 1
    if agent inspect block forward ≠ then
      say "Found the stone!"
      agent destroy forward
      agent collect all
    else
      say "No stone here!"
    agent move up by 1
```

Correct Code:

```
on chat command "stone"
  repeat 4 times
  do
    agent move left by 4
    agent destroy down
    agent move down by 1
    if agent inspect block forward = then
      say "Found the stone!"
      agent destroy forward
      agent collect all
    else
      say "No stone here!"
    agent move up by 1
```

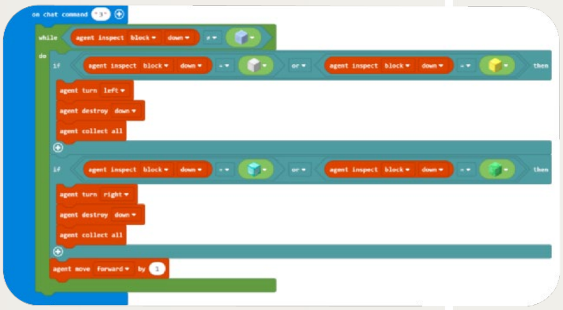
9. Coding Fundamentals 9

Block 2, Lesson 4 – Ganymede (Jupiter's Moon) & Saturn's Rings: Comparisons and Conditionals

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

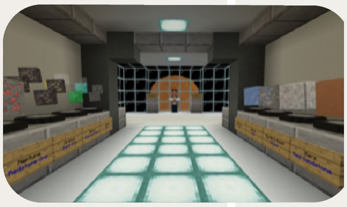
- Comparisons – Boolean (logic) Operators
- Boolean Operators with Conditionals

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/8948529193526873-Coding%20Fundamentals_Block%202%20Lesson%204_Ganymede%20and%20Saturns%20Rings.pdf



```
on chat command "3"
  while agent inspect block down ≠
  do
    if agent inspect block down = or agent inspect block down = then
      agent turn left
      agent destroy down
      agent collect all
    if agent inspect block down = or agent inspect block down = then
      agent turn right
      agent destroy down
      agent collect all
    agent move forward by 1
```

10. *Coding Fundamentals 10*



Block 2, Lesson 6 – Revisiting Planets: Summative Collaborative Build Task

MEE CF https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals

Concept Review

Split students into teams of 3-4 students. Some useful tips can be found here: https://education.minecraft.net/wp-content/uploads/Minecraft-Education-Edition-Multiplayer-Guide-1.pdf

There are 4 areas to choose from:

- Venus
- Mars
- Ganymede (Jupiter's Moon)
- Mercury

Each area has coding activities that involve:

- Conditional
- Repeat Loop
- While Loop
- Debugging
- Comparison Operator
- Boolean

Students will need to work together on the following types of activities:

- Brainstorming
- Planning
- Building
- Testing
- Recording

Download the educator guide here – https://education.minecraft.net/lessonsupportfiles/5459817952550881-Coding%20Fundamentals_Block%202%20Lesson%206_Revisiting%20Planets.pdf

Supporting Content

- **MEE CF,** Coding Fundamentals. Blocks 1, 2 and 3, lessons 1-24 https://education.minecraft.net/en-us/resources/computer-science-subject-kit/coding-fundamentals
- **MEE CodWM,** Coding With Minecraft, Units 1-10, 1st unit – https://education.minecraft.net/en-us/lessons/coding-introduction
- **MC CSP AP** 3&5, Computer Science Principles course, Units 3 & 5 Algorithms and Programming Part 1 & 2, https://microsoft.github.io/makecode-csp/unit-3/algorithms-programming & https://microsoft.github.io/makecode-csp/unit-5/algorithms-programming
- **MC CS** Intro 1 – 3, Beta, Variables, Sprites, Motion, Loops, 1st unit – https://arcade.makecode.com/courses/csintro
- Clever Computers – https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/clever-computers
- https://kids.britannica.com/kids/article/computer/352990
- Bebras International Challenge https://www.bebras.org/

How Can Code Help Make Predictions?

Domain – Coding

Big Idea – Writing Programs

Level – 9

Time allocation – 10 hours

Learning Goals

Main

- Develop programs that combine control structures, including nested loops and compound conditionals, CSTA 2-AP-12
- Create interactive data visualizations using software tools to help others better understand real-world phenomena, CTSA 3A-DA-11
- Use algorithms to make predictions by turning probability into code, MCSG

Supplementary

Understand the difference between classification and regression

Concepts

- Mathematical methods can be converted to code and used to solve problems
- Decision trees
- Regression analysis

Challenges

Computational Thinking

- Build a decision tree to predict a categorisation
- Predict fraud from credit card data
- Build a Monty Hall game simulation with code
- Predict where an object may fall on the ground from a moving aircraft

Data Literacy – Analyse climate data with Azure Notebooks

Design Thinking – Design user interactions for the computational thinking challenges

Example Lesson Outlines

1. Probability

Minecraft EE, What's The Probability?

Here, students will:

- Play in survival mode to collect items from certain mobs or blocks
- Enter data into a spreadsheet
- Calculate their personal 'drop rates' in terms of decimal probability and a percentage
- Discover the probability of items dropped

MEE Prob, https://education.minecraft.net/en-us/lessons/whats-the-probability

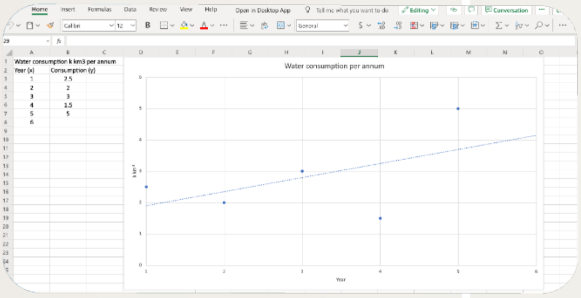
2. Introduction to Linear Regression – Excel

Making prediction using code usually involves either regression or classification. Both regression and classification can be understood first through practical exercises in Excel, and from here students can apply their knowledge of regression and classification in Python.

Let's first start with Linear Regression in Excel. First, enable students to understand how Linear regression works by pointing them to this link – Reg Mod, https://docs.microsoft.com/en-gb/training/modules/train-evaluate-regression-models/2-what-is-regression

Using the data in the table below, get students to predict the value for Year 6.

Water consumption k km3 per annum	
Year (x)	Consumption (y)
1	2.5
2	2
3	3
4	1.5
5	5
6	?

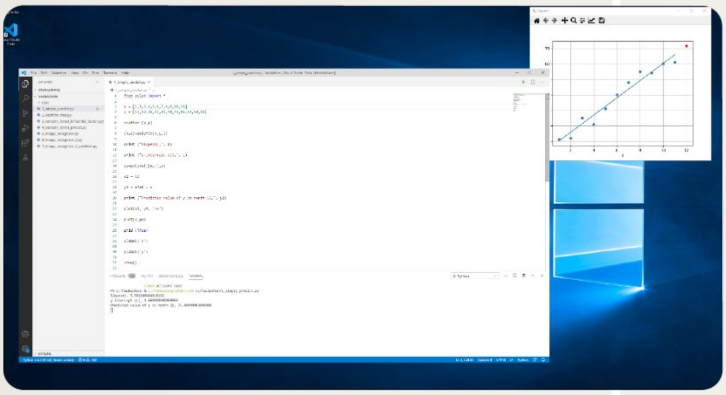


Water consumption k km3 per annum	
Year (x)	Consumption (y)
1	2.5
2	2
3	3
4	1.5
5	5
6	

3. *Linear Regression in Python*

Once they understand the concept students can go on to coding a Linear Regression model in Python. For example:

```
from pylab import *
x = [1,2,3,4,5,6,7,8,9,10,11]
y = [11,12,25,21,31,40,48,55,54,60,61]
scatter (x,y)
(m,c)=polyfit(x,y,1)
print ("Slope(m),", m)
print ("y-intercept (c),", c)
yp=polyval([m,c],x)
x2 = 12
y2 = m*x2 + c
print ("Predicted value of y in month 12,", y2)
plot(x2, y2, 'ro')
plot(x,yp)
grid (True)
xlabel('x')
ylabel('y')
show()
```



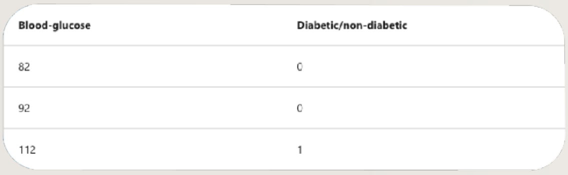
Either give students no code or jumbled code and get them to change the data to change outcomes.

4. *Logistic Regression in Excel*

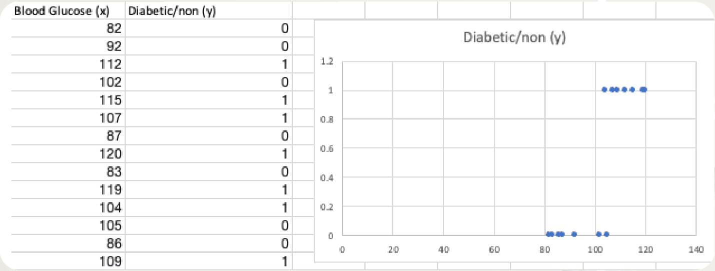
Classification is another essential tool for making predications with code and data. Logistic Regression is a commonly used tool and provides a good introduction to classification.

First, students need to understand how Logistic Regression works, so point them to this resource – Log Reg, https://docs.microsoft.com/en-us/training/modules/introduction-classification-models/2-what-is-classification

Once they understand the basic concept, get students to reproduce the classification shown here in Excel, using the patient data given in the table –



Blood-glucose	Diabetic/non-diabetic
82	0
92	0
112	1



Blood Glucose (x)	Diabetic/non (y)
82	0
92	0
112	1
102	0
115	1
107	1
87	0
120	1
83	0
119	1
104	1
105	0
86	0
109	1

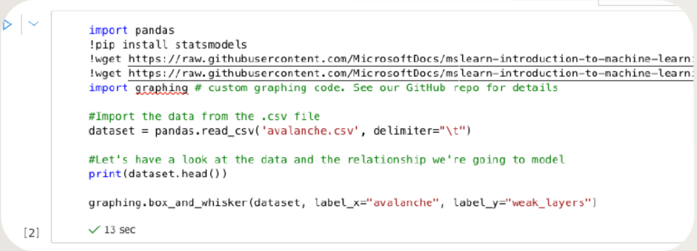
Note, it's sufficient for students to just plot the data points to see the level of blood glucose above or below which a patient is diabetic or not.

5. Logistic Regression in Python 1

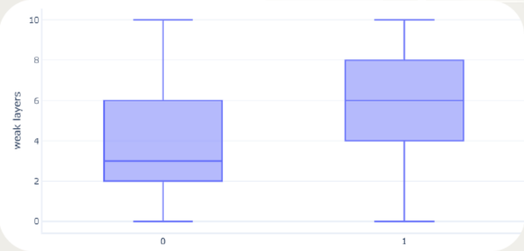
Having learned the basics of Logistic Regression using Excel, students can now go on to build, assess and improve a Logistic Regression model in Python. This can be done in 2 lessons – the first is about building and assessing the model, the second will focus on improving it.

Topics covered

- What are classification models?
- Exercise – Build a simple logistic regression model
- Assessing a classification model
- Exercise – Assessing a logistic regression model



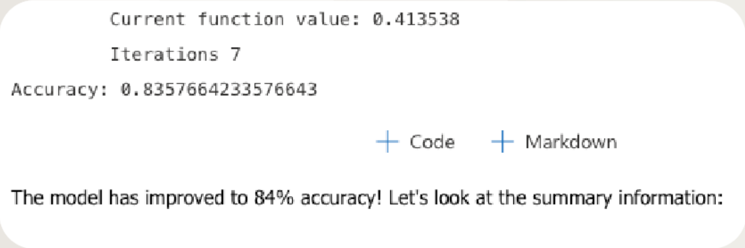
```
import pandas
!pip install statsmodels
!wget https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to-machine-learn:
!wget https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to-machine-learn:
import graphing # custom graphing code. See our GitHub repo for details

#Import the data from the .csv file
dataset = pandas.read_csv('avalanche.csv', delimiter="\t")

#Let's have a look at the data and the relationship we're going to model
print(dataset.head())

graphing.box_and_whisker(dataset, label_x="avalanche", label_y="weak_layers")
```



ML CLS-ML, https://docs.microsoft.com/en-us/training/modules/understand-classification-machine-learning/

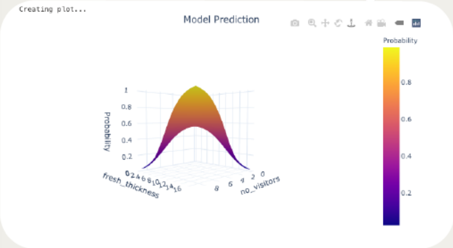
6. Logistic Regression in Python 2

The model built in lesson 5 will be wrong around 40% of the time, so this lesson focusses on improving and visualising the model.



```
         Current function value: 0.413538
         Iterations 7
Accuracy: 0.8357664233576643
```

The model has improved to 84% accuracy! Let's look at the summary information:



ML CLS-ML, https://docs.microsoft.com/en-us/training/modules/understand-classification-machine-learning/

7. Predict Rocket Launch Delays with Machine Learning 1

This Microsoft Learn learning path introduces students to the world of machine learning – using code to make predictions. Students will take a real-life problem that NASA faces and apply machine learning to solve it.

There are three lessons in this learning path, the first of which is an introduction to how NASA chooses a date for a rocket launch.

Introduction to Rocket Launches

Learning objectives

In this lesson, students will learn the main concepts behind making predictions related to launching a rocket:

- The challenges weather can pose for a rocket launch
- The data science lifecycle
- How machine learning works
- The role ethics play in machine learning

Topics covered

- Data to predict weather years in advance
- Launch day weather analysis
- Machine learning and the data science lifecycle
- Set a goal and get expertise
- Collect, clean, and manipulate data
- Choose an algorithm and train and test your model
- Deploy your machine learning model
- How humans and machine learning models learn
- Ethics in data science and machine learning

ML NASA PLD, https://docs.microsoft.com/en-us/training/paths/machine-learning-predict-launch-delay-nasa/

https://www.youtube.com/watch?v=eK3ATduvFGM

8. *Predict Rocket Launch Delays with Machine Learning 2*

Data Collection and Manipulation

To determine the best time to launch a rocket into space NASA collects and processes weather and other data in machine learning models. Using this Microsoft Learn unit, students will learn about the steps to import data into Python and how to clean the data for use in creating machine learning models.

Learning objectives

In this module, students will conduct practical coding work to:

- Explore weather data on days crewed and uncrewed rockets were launched
- Explore weather data on the days surrounding launch days
- Clean the data in preparation for training the machine learning model

Photo by NASA on Unsplash

Topics covered

- Determine the rocket launch questions to ask
- Explore the rocket launch data to gain an understanding
- Import Python libraries and rocket launch data
- Clean weather data to analyze rocket launch criteria
- Consider additional data to include

ML NASA PY, https://docs.microsoft.com/en-us/training/modules/collect-manipulate-data-python-nasa/

9. *Predict Rocket Launch Delays with Machine Learning 3*

Build a Machine Learning Model

In this lesson students focus on a local analysis of their data by using scikit-learn and use a decision tree classifier to gain knowledge from raw weather and rocket launch data.

Learning objectives

- The importance of column choosing.
- How to split data to effectively train and test a machine learning algorithm.
- How to train, test, and score a machine learning algorithm.
- How to visualize a tree classification model.

10. *Skills and knowledge check*

Supporting Content

- **Microsoft Learn for Students, ML4S FtP, First-time Python**, https://docs.microsoft.com/en-us/learn/modules/intro-to-python/

How Can We Use Software Against a Pandemic?

Domain – Coding

Big Idea – Writing Programs

Level – 10

SDG – 3. Good Health and Well-Being

Time allocation – 10 hours

Learning Goals

Main

- Use data analysis tools and techniques to identify patterns in data representing complex systems, CSTA 3B-DA-05
- Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects, CSTA 3A-AP-17
- Access pre-existing functionality from standard libraries, MCSG

Supplementary

Create a computational model that represents the relationships among different elements of data collected from a phenomenon or process

Concepts

- Understanding mathematics, especially statistics, is a key part of Computer Science, and often coding involves turning math into code.
- Governments need to have reliable models to make policy decisions to control infectious disease.
- In epidemiology (the study of infectious diseases) a SIR model allows the user to model the progress of an epidemic over time using parameters for susceptibility, infection, and recovery

- Modular programming is a software technique that separates functionality into independent units to promote code reusability and extensibility
- Business Intelligence software such as Power BI and Tableau allows the user to interactively create reports and visualizations from data
- A web service serves a program that processes inputs from one program into output used by another and is a building block of web applications

Challenges

Computational Thinking – Write an SIR model program in Python as part of a pandemic control dashboard web service

Data Literacy – Create models that accepts user inputs parameters and predicts the progress of a pandemic through time

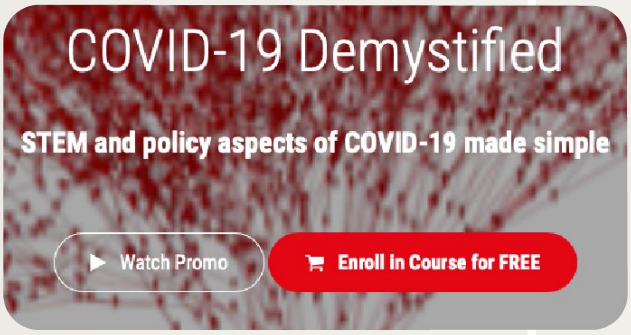
Design Thinking – Design and build reports in Power BI to visualize the spread of a disease under different scenarios

Example Lab Outlines

10 labs, 1 hour each

1. Data and Models

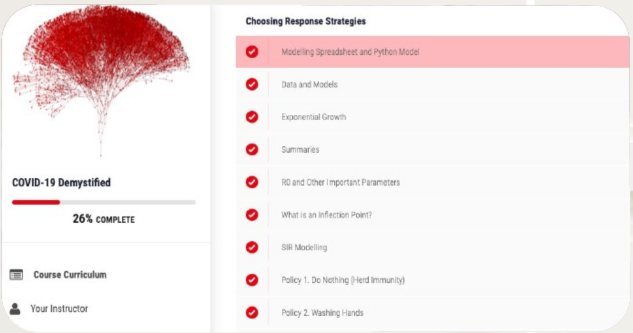
These labs are based on an a free COVID-19 Demystified course and students should enrol on this to access the learning content – https://www.learn-tech.courses/p/covid-19-demystified



COVID-19 Demystified

STEM and policy aspects of COVID-19 made simple

Students should use the "Choosing Response Strategies" module which they will work in for the first 5 labs.



COVID-19 Demystified

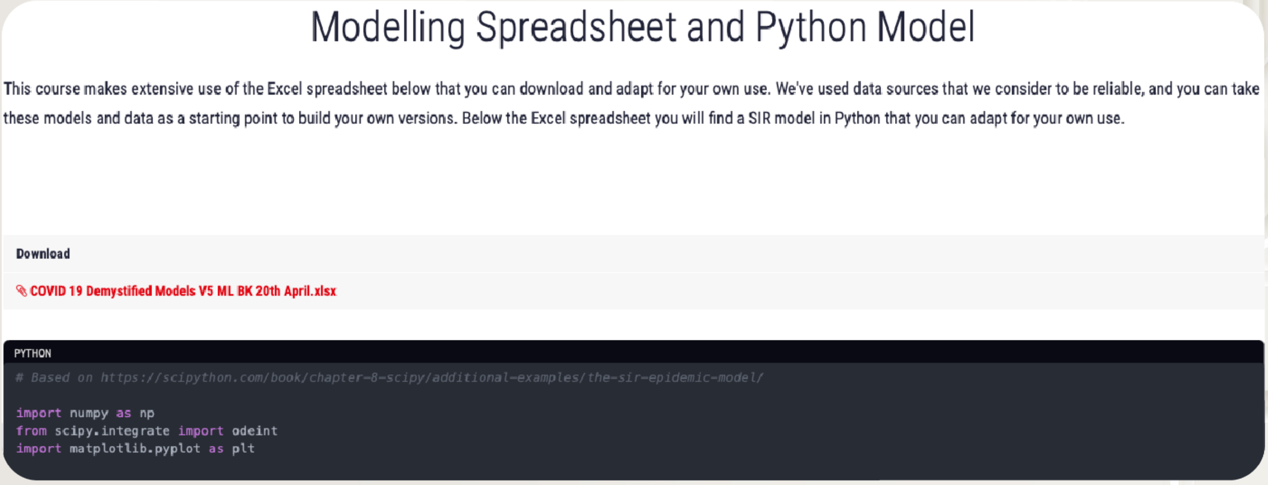
26% COMPLETE

Choosing Response Strategies

- ✓ Modelling Spreadsheet and Python Model
- ✓ Data and Models
- ✓ Exponential Growth
- ✓ Summaries
- ✓ R0 and Other Important Parameters
- ✓ What is an Inflection Point?
- ✓ SIR Modelling
- ✓ Policy 1. Do Nothing (Herd Immunity)
- ✓ Policy 2. Washing Hands

These first 5 labs are focussed on the background to the task that students will be set in the last 5 labs, and whilst the material explores statistics and data representations, these are critical for subsequent coding and product development work.

After familiarising themselves with the rest of the course, they should explore at the Excel modelling spreadsheet and Python model which form the basis for the rest of this module.



Modelling Spreadsheet and Python Model

This course makes extensive use of the Excel spreadsheet below that you can download and adapt for your own use. We've used data sources that we consider to be reliable, and you can take these models and data as a starting point to build your own versions. Below the Excel spreadsheet you will find a SIR model in Python that you can adapt for your own use.

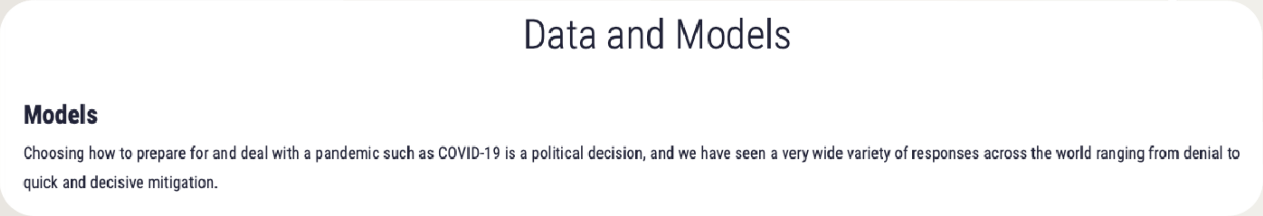
Download

COVID 19 Demystified Models V5 ML BK 20th April.xlsx

PYTHON

```
# Based on https://scipython.com/book/chapter-8-scipy/additional-examples/the-sir-epidemic-model/

import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
```

After this, they should work their way through the Data and Models section until they are familiar with the basis on which the rest of this module is built.



Data and Models

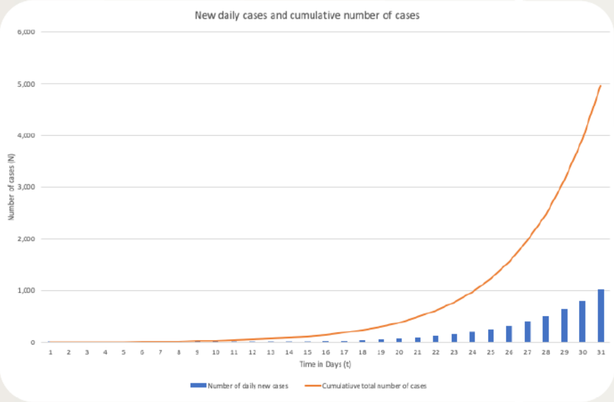
Models

Choosing how to prepare for and deal with a pandemic such as COVID-19 is a political decision, and we have seen a very wide variety of responses across the world ranging from denial to quick and decisive mitigation.

2. *Exponential Growth & Summaries*

We know that pandemics grow if policies are not put in place, but what kinds of growth patterns can we model? What kind of summaries of trends do governments need to see in order to keep track of development and make decisions?

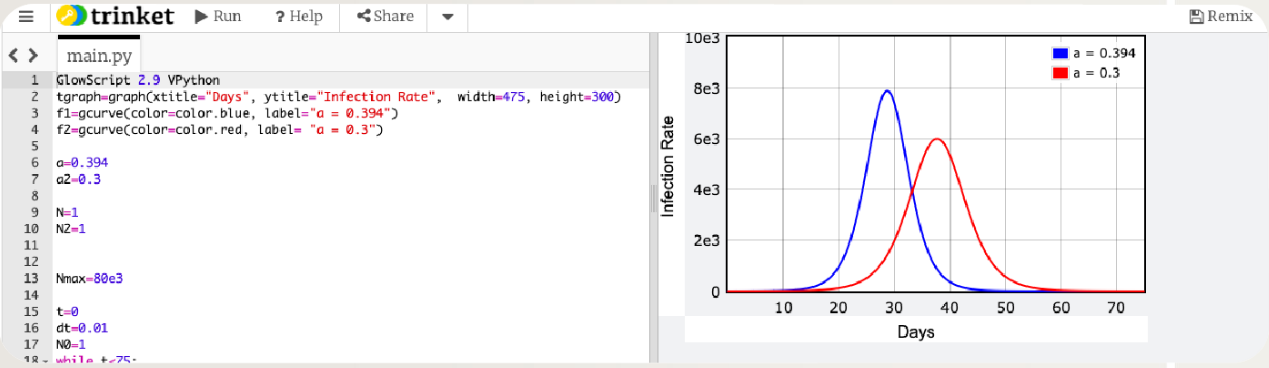
In these two modules students are encouraged to change the data to see different patterns and understand how the visualisations are constructed.



3. *R$_0$, Important Parameters, Inflection Points*

Why is R$_0$ such an important number to know, and how do epidemiologists calculate it? What is meant by “inflection point”, and why is it important?

Students should gain a full understanding of R$_0$ and inflection points, then change the parameters of the Python models to see different results.



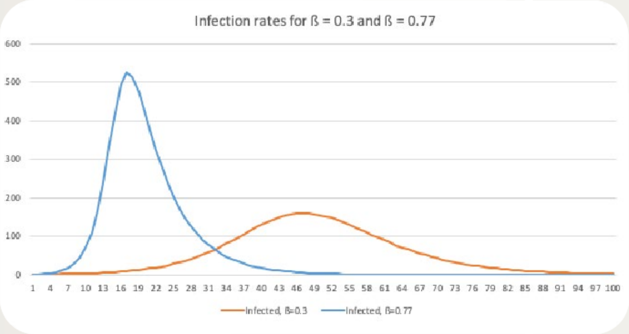
```
GlowScript 2.9 VPython
tgraph=graph(xtitle="Days", ytitle="Infection Rate",  width=475, height=300)
f1=gcurve(color=color.blue, label="a = 0.394")
f2=gcurve(color=color.red, label= "a = 0.3")

a=0.394
a2=0.3

N=1
N2=1


Nmax=80e3

t=0
dt=0.01
N0=1
while t<75:
```

If time permits, students are also encouraged to find ways to take the same code used here and make it run in different environments.

4. *SIR Modelling*

SIR is a powerful and widely used model that can predict the number of people who will become infected. SIR stands for

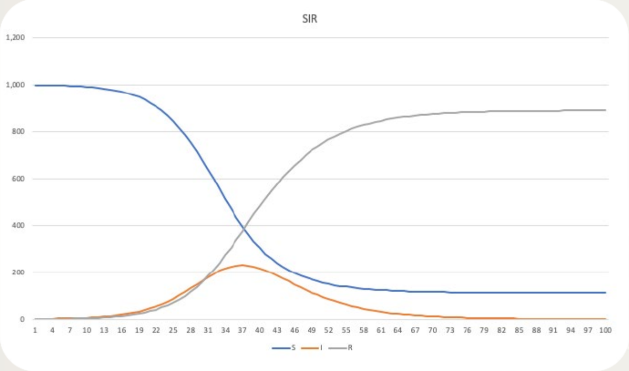
- Susceptible to infection
- Infected
- Recovered or removed



Students are encouraged here to try different inputs into the model in both Excel and Python to compare results.

5. *Modelling Different Policies*

To understand the options and dilemmas behind political decision making, we need to understand the different effects that different approaches are likely to have. For this we need to run various models.



Here students will use a SIR model to evaluate the following 3 policy scenarios:

- Do nothing
- Widespread hand washing
- Social distancing

If time permits, students can also model 'wearing masks'.

6. *Design & Build a Pandemic Control Dashboard 1*

With an understanding of the kinds of modelling that needs to be done to deal with a pandemic, students' attention should now be drawn to the following design brief which should be undertaken in teams of between 3 and 5 students over the next 4 labs –

> *"Design and build a prototype web service and dashboard that senior decision makers in government can use to help make decisions that are effective in dealing with a pandemic. The prototype should include simple visualisations showing trends, ratios and key numbers and should be work on both mobile phone and PC interfaces".*

The first part of this mini design and build project is for students to decide what information needs to be presented to senior decision makers. Once this is decided they should work through the following Microsoft Learn learning path, and use it to guide their product development –

Consume data with Power BI – **ML CDPBI**, https://docs.microsoft.com/en-us/training/paths/consume-data-with-power-bi/

Explore what Power BI can do for you – **ML EPBI**, https://docs.microsoft.com/en-us/training/modules/explore-power-bi-service/

- Understand Power BI service basic concepts and terminology.
- Find your content in dashboards, reports, and apps.
- View and export data from dashboards and reports.
- View filters that are used in a report.
- Explain the relationship between dashboards and reports, visualizations, and tiles.
- Display action menus for tiles and details for report visualizations.
- Collapse and expand the nav pane and the Filters and Bookmarks panes.
- Subscribe to a dashboard and report.
- Set a data alert.
- Collaborate by using dashboard commenting.

7. Design & Build a Pandemic Control Dashboard 2

Analyze data with Power BI – **ML ADPBI**, https://docs.microsoft.com/en-us/training/modules/analyze-data-power-bi/

- Explore how filters are applied to your data
- Learn about different ways that you can filter reports
- Create personal bookmarks
- Analyze and export data

8. Design & Build a Pandemic Control Dashboard 3

Get to your data on the go with the Power BI Mobile apps – **ML EPBIA**, https://docs.microsoft.com/en-us/training/modules/explore-power-bi-mobile-apps/

- Find your content
- View and interact with your data
- Access and use filters and bookmarks
- Share reports and dashboards with colleagues
- Set data alerts
- Create shortcuts to reports and dashboards
- Use the device's camera and location capabilities to get data from out in the real world
- Configure app settings to personalize your experience with the app.

9. Design & Build a Pandemic Control Dashboard 4

In this lab, students should pull together all the components needed to build, present, and evaluate the final product.

https://docs.microsoft.com/en-us/power-bi/consumer/mobile/mobile-apps-for-mobile-devices

10. Skills & knowledge check

Assessment of the mini design and build project is via group evaluation of projects, and teacher assessment of contributions to the team by individual members.

Individual students should also return to the COVID-19 Demystified course and work through the following -

- Exercise 5, Modelling
- Checkpoint 5

Supporting Content

- **ML4S VPBI,** Use visuals in Power BI, https://docs.microsoft.com/en-gb/learn/modules/visuals-in-power-bi/
- **PBI,** Power BI, https://docs.microsoft.com/en-us/power-bi/guided-learning/
- **ML4S 11**, Import standard library modules – https://docs.microsoft.com/en-us/learn/modules/python-standard-library/

How Can Code Reduce Hospital Waiting Lists?

Domain – Coding

Big Idea – Writing Programs

Level – 12

SDG – 3. Good Health and Well-Being

Time allocation – 6:40

Learning Goals

Main

- Understand the power of OOP for modelling real-world phenomena, create a DRY OO program with high modularity and extensibility, MCSG
- Explain what a relational database is, and understand the benefits of storing data in multiple tables, CAS D&DR 8

Supplementary

- Use object-oriented programming in a modern high-level language such as C, C++ or C# or Python to represent and solve real-world problems
- Demonstrate the capability to design and organize an object-oriented program for reusability, extensibility, and robustness

Concepts

- Object-Oriented Programming (OOP) is a method of organizing a program around objects rather than functions
- The key differences between OOP and traditional procedural programming are encapsulation, inheritance, and polymorphism
- A key goal of OOP design is DRY (do not repeat yourself) as this promotes program's maintainability which ensures robustness for enterprise use
- UML (Unified Modelling Language) is used to map out how the program objects behave and relate, providing a visual blueprint for the program.

Challenges

Computational Thinking – Write a program to allocate patients and surgeons to surgeries based on matching schedules

Data Literacy – Summarize your program results by person/time/role and other dimensions, visualizing the output appropriately

Design Thinking – Design a program structure in terms of inputs, modules to perform operations and desired output

Project Framework

Phase 1, Frame The Problem – 40 Mins

Brief – Surgery Backlog Management Software

Whilst COVID-19 may be thought of as 'in the past', its past and current impact has lead to a backlog of patient surgeries which could last years. Health systems need to calculate how best to prioritise patient flow, and how to best treat those waiting for surgery while the backlog is being reduced.

Your brief here is to build a rough prototype of a surgery backlog management software solution that optimises hospital time whilst managing the risk to patients of waiting for procedures. Key principles include:

- Every life is of equal value
- Hospital capacities cannot grow quickly enough to meet current demand
- The risk of patient harm through waiting for surgery has to be balanced against the reality of insufficient hospital capacity to treat everyone in the way that they need.

The first task on this project is to frame the problem in terms of the input data and type of calculation that would need to be made to reduce waiting lists.

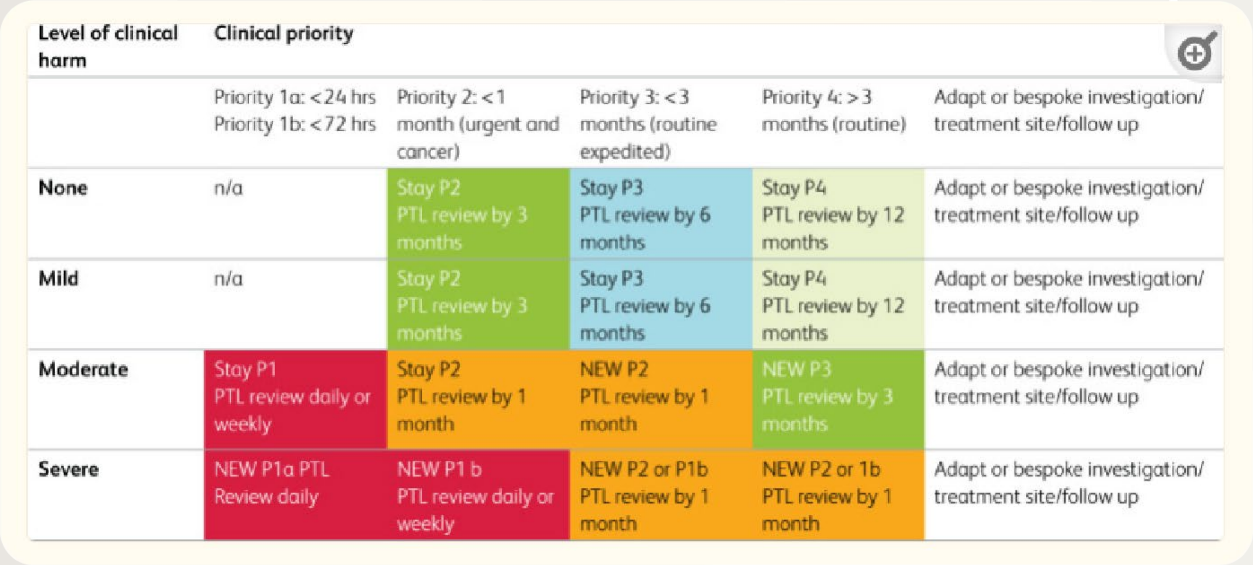
Key questions –

- How should patients be prioritised for hospital treatment?
- How can patients be treated most effectively given the constraints?
- What should happen to patients whilst they wait?
- How can data best be obtained and used?
- What kinds of calculations does waiting list management software need to perform?
- What information needs to be presented to the health service decision makers who decide what happens to patients?

Part 1 – Prioritisation

Whilst not endorsing this approach, understanding the thinking behind an attempt to build a standardised method of patient prioritisation developed by the The Royal College of Surgeons of England, and reported in the Royal College of Physicians "Future Healthcare Journal", could be a good way to start framing this project.

The Royal College of Physicians' 'standardised matrix', maps priority categories against the clinical harm of waiting for surgery.



Level of clinical harm	Clinical priority				
	Priority 1a: <24 hrs Priority 1b: <72 hrs	Priority 2: <1 month (urgent and cancer)	Priority 3: <3 months (routine expedited)	Priority 4: >3 months (routine)	Adapt or bespoke investigation/ treatment site/follow up
None	n/a	Stay P2 PTL review by 3 months	Stay P3 PTL review by 6 months	Stay P4 PTL review by 12 months	Adapt or bespoke investigation/ treatment site/follow up
Mild	n/a	Stay P2 PTL review by 3 months	Stay P3 PTL review by 6 months	Stay P4 PTL review by 12 months	Adapt or bespoke investigation/ treatment site/follow up
Moderate	Stay P1 PTL review daily or weekly	Stay P2 PTL review by 1 month	NEW P2 PTL review by 1 month	NEW P3 PTL review by 3 months	Adapt or bespoke investigation/ treatment site/follow up
Severe	NEW P1a PTL Review daily	NEW P1 b PTL review daily or weekly	NEW P2 or P1b PTL review by 1 month	NEW P2 or 1b PTL review by 1 month	Adapt or bespoke investigation/ treatment site/follow up

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8004326/

If students choose to use this matrix, their next task is to work out how to take an example patient use case and work out which priority level the patient belongs to using code. Here, students need to work out what data needs to be collected and analysed, and what kind of calculations are needed to categorise patients accurately. It's important to collect the optimal amount of data about the patient – too much data becomes unmanageable, and calculations based a lot of different features isn't necessarily going to be more accurate than the alternative.

Part 2 – Alternative Treatments

Those patients who are not deemed highest priority will need to be given an alternative to hospital admission and additional diagnosis to ensure that their prioritisation doesn't change. For example, one of the most common operations is for cataracts, so what are the alternatives to having patients come into hospital for these? Some other examples of how innovation and new technology can help reduce hospital waiting lists can be found here, https://www.gov.uk/government/news/innovation-and-new-technology-to-help-reduce-nhs-waiting-lists

Alternative Approach to The Brief

Students could focus on an analysis of a subset of clinical diagnostics as set out here: ahttps://www.england.nhs.uk/coronavirus/documents/clinical-prioritisation-of-waiting-lists-for-endoscopy-and-diagnostic-procedures-framework-and-guidance/ – note Appendix A, for an alternative categorisation of patient prioritisation.

Students taking this pathway would also need to specify an alternative to hospital treatment for those who are not considered highest priority.

Phase 2, Take Your First Steps With C# – 5 hours

Whilst other object orientated languages could be suitable for this project, students can design and build their solution here using C#. These guides to C# can be useful to those developing in any OOP language –

- Take your first steps with **C#, ML C#1st**, https://docs.microsoft.com/en-us/training/paths/csharp-first-steps/
- Write your first C# code, **ML4S C#1**, https://docs.microsoft.com/en-us/learn/modules/csharp-write-first/

Phase 3, Present and Evaluate – 1 hour

Assessment of this project is via group evaluation of projects, and teacher assessment of contributions to the team by individual members.

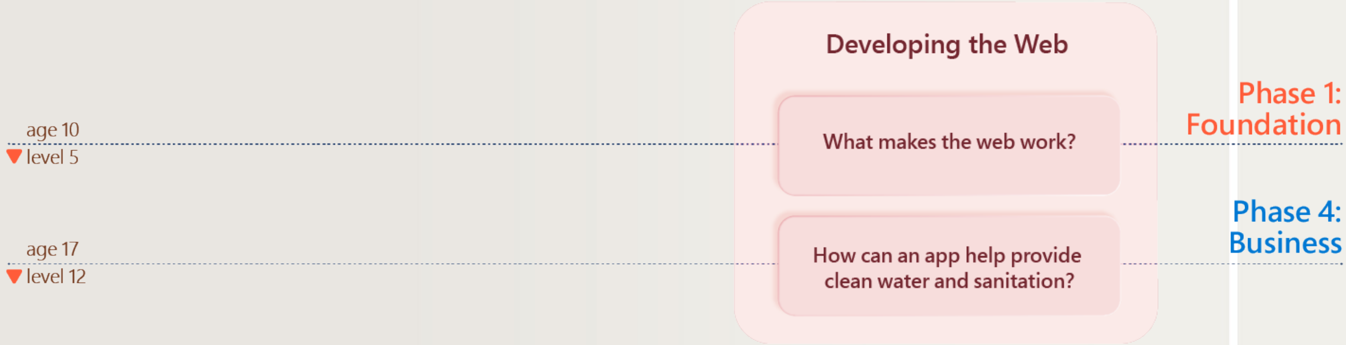
Each team, ideally 5 diverse members of mixed ability, is given 5 mins to present their solution in slides and a demo, followed by 5 min Q&A. All other students in the class are encouraged to ask questions and one way to do this is to award team points for good questions.

Note that this presentation and evaluation session is 40 mins shorter than the previous lab (How Can We Build Circular Economies?), so ideally, each team will make a 5 min video of their solution as they develop and finish which can help with timekeeping at the group Present and Evaluate session.

Supporting Content –

- **ML4S PO**, Python Object Oriented Programming, https://docs.microsoft.com/en-gb/learn/modules/python-object-oriented-programming/
- **ML4S DFS**, Distributed file systems, https://docs.microsoft.com/en-us/learn/modules/cmu-case-study-distributed-file-systems/
- **ML4S WAT**, Write automated tests, https://docs.microsoft.com/en-us/learn/modules/rust-automated-tests/
- **CE**, How to use character encoding classes in .NET, https://docs.microsoft.com/en-us/dotnet/standard/base-types/character-encoding
- **CS4FN** programming php – http://www.cs4fn.org/fundamentals/programming.php

Developing the Web



Here, students will get the opportunity to learn how the apps they use on their phones work, and develop skills needed to make the World Wide Web work for them, by learning the skills necessary to build web pages, sites and services as well as developing an understanding of how the Web itself functions.

What Makes the Web Work?

Domain – Coding

Idea – Developing the Web

Level – 5

Time allocation – 10 hours

Learning Goals

Main

Construct static web pages using HTML and CSS, CAS C&N 5

Supplementary

- Appreciate that the World Wide Web is a decentralized web with links between sites and pages
- Understand HTML as the foundation of web pages
- Acquire the capability to create a webpage using HTML, CSS and JavaScript

Concepts

- World Wide Web
- HTML, CSS and JavaScript
- Hyperlinks

Challenges

Computational Thinking – Build a navigable web page with information and pictures about your neighbourhood (hyperlink between sections)

Data Literacy – Map desired web page features to HTML tags

Design Thinking – Design a web page to share what you love about your neighbourhood

Example Lesson Outlines

For web development, understanding JavaScript, HTML and CSS (Cascading Style Sheets) are essential skills, and this module is designed to

The first 6 lessons focus on JavaScript and will develop skills – such as arrays and loops – which are transferable across multiple languages. Lesson 7 introduces HTML and CSS, and lessons 8, 9 and 10 are about students building their own web pages.

Students can follow 'the Microsoft Learn Web Development for Beginners' program – ML4S WDFB, https://learn.microsoft.com/en-us/training/paths/web-development-101/

1. Introduction to Programming

Consume data with Power BI – **ML CDPBI**, https://docs.microsoft.com/en-us/training/paths/consume-data-with-power-bi/

Learn about programming languages and the tools that developers use to create software, with a focus on browsers.

ML4S I2P, https://learn.microsoft.com/en-us/training/modules/web-development-101-introduction-programming/

2. Learn the Basics of Web Accessibility

In this module, students will learn how to design web pages are accessible to people with impairments:

- Learn about tools that users use to browse through webpages
- Learn about tools that developers use to ensure that web pages are accessible
- Gain skills for ensuring that the web pages they create are accessible

ML WACC, https://learn.microsoft.com/en-us/training/modules/web-development-101-accessibility/

3. JavaScript Variables and Data Types

Learn about variables and the data types that populate them in the context of JavaScript.

ML Var, https://learn.microsoft.com/en-us/training/modules/web-development-101-variables/

4. Create Modular Code by Using Functions in JavaScript

Functions are the building blocks for any application students will create. In this lesson students will create named, reusable sections of code to help make it more readable and maintainable.

ML Func, https://learn.microsoft.com/en-us/training/modules/web-development-101-functions/

5. Make Decisions with JavaScript

In this lesson, students will learn how comparing variables, using Booleans, and using if/else statements allow them to make decisions in their code.

ML Bool, https://learn.microsoft.com/en-us/training/modules/web-development-101-if-else/

6. JavaScript Arrays and Loops

In this lesson, students will learn how comparing variables, using Booleans, and using if/else Learn how to manipulate and store data within JavaScript.

ML Array, https://learn.microsoft.com/en-us/training/modules/web-development-101-arrays/

7. Build a simple website using HTML, CSS, and JavaScript, Part 1

In this lesson, students will learn how to:

- Create a basic web page using HTML
- Apply styles to page elements using CSS
- Create themes using CSS
- Add support for switching between themes using JavaScript
- Inspect the website using browser developer tools

https://learn.microsoft.com/en-gb/training/modules/build-simple-website/

8. *Build a simple website using HTML, CSS, and JavaScript, Part 2*

In this lesson, students will continue learn how to create basic web pages using HTML, CSS and JavaScript.

Brief

Students should build web pages that explain the UN's Sustainable Development Goals (SDGs).

ML WebJS, https://learn.microsoft.com/en-gb/training/modules/build-simple-website/

9. *Build a simple website using HTML, CSS, and JavaScript, Part 2*

In this lesson, students will continue to build basic web pages using HTML, CSS and JavaScript.

ML WebJS, https://learn.microsoft.com/en-gb/training/modules/build-simple-website/

10. *Build a simple website using HTML, CSS, and JavaScript, Part 3*

In this lesson, students will complete and show their basic web pages.

ML WebJS, https://learn.microsoft.com/en-gb/training/modules/build-simple-website/

Supporting Content

- Create and publish a static web app with Gatsby and Azure Static Web Apps, **ML Gats**, https://learn.microsoft.com/en-us/training/modules/create-deploy-static-webapp-gatsby-app-service/
- Create and host web sites by using GitHub Pages, **ML4S WSGH**, https://docs.microsoft.com/en-us/learn/modules/create-host-web-sites-github-pages/
- Code.org (CSD Unit 2), Web Development. HTML, CSS, build a personal portfolio website – https://studio.code.org/s/csd2-2019
  https://websitesetup.org/
  https://techdifferences.com/difference-between-html-and-css.html
  https://www.youtube.com/watch?v=PlxWf493en4
  https://www.w3schools.com/html/default.asp
  http://www.webyurt.com/popular-types-of-websites
  https://kids.britannica.com/students/article/World-Wide-Web-WWW/571030
  https://webfoundation.org/about/vision/history-of-the-web/
  https://kids.britannica.com/kids/article/Internet/353293
  http://www.kidscodeprojects.com/HTML/html.html

How Can an App Help Provide Clean Water and Sanitation?

Domain – Coding

Big Idea – Developing the Web

Level – 12

SDG – 6. Clean Water and Sanitation

Time allocation – 6:40 hours

Learning Goals

Main

Apply JavaScript to create responsive, interactive webpages – Computational Thinking, MCSG

Supplementary

- Use a server-side scripting language to read from and write to the database from the application
- Program SQL queries to read from and write to a database

Concepts

- A modern web application consists of three tiers: 1. Web Tier; 2. Application Tier and 3. Storage Tier
- JavaScript (web tier) allows the programming of complex, interactive features on a website
- A server-side scripting language such as ASP.net, node.js or Python (application tier) allows the web tier to interact with the database and fetch or put data
- An RDBMS (storage tier) is a relational database that persists application data in a structured, relational form accessible by SQL (Structured Query Language)

Challenges

Computational Thinking – Write a web application to input, store and display water quality information

Data Literacy – Organize data into a relational model for update and querying

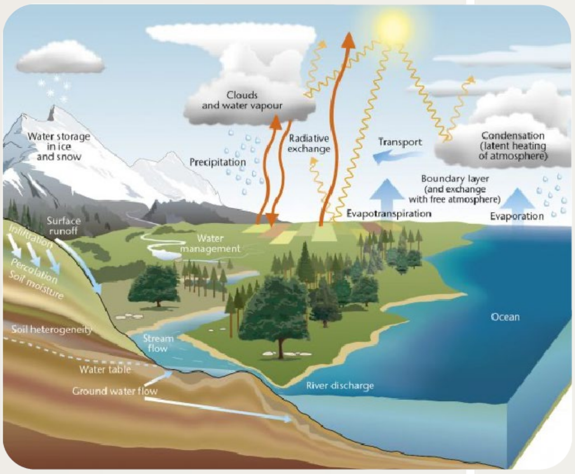
Design Thinking – Design a web application, diagramming its architecture and functionality

Example Project Outline

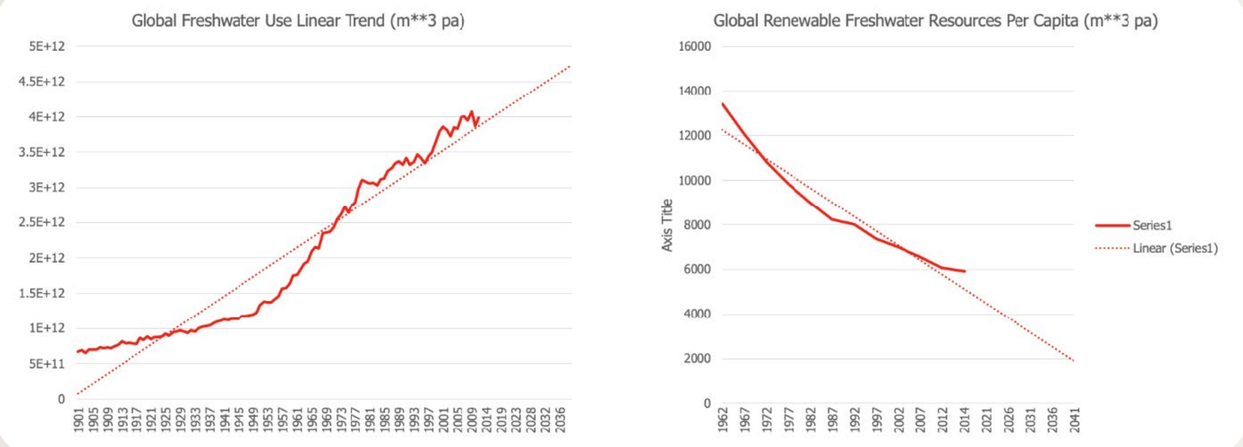
Climate change impacts the world's water in complex ways. Consider a water cycle diagram, like the one below; global warming is altering nearly every stage in the cycle. These changes will put pressure on drinking water supplies, food production, property values, and more all around the world[lv].

As heat increases so does flooding and drought. Flooding can contaminate fresh water[lvi].

The World Economic Forum (WEF) states that clean water scarcity is one of the greatest challenges of our time. Global use of that freshwater has been growing at roughly twice the rate of the global population for the past century.



Climate change's impact on fresh water.
Image source, UK Met Office



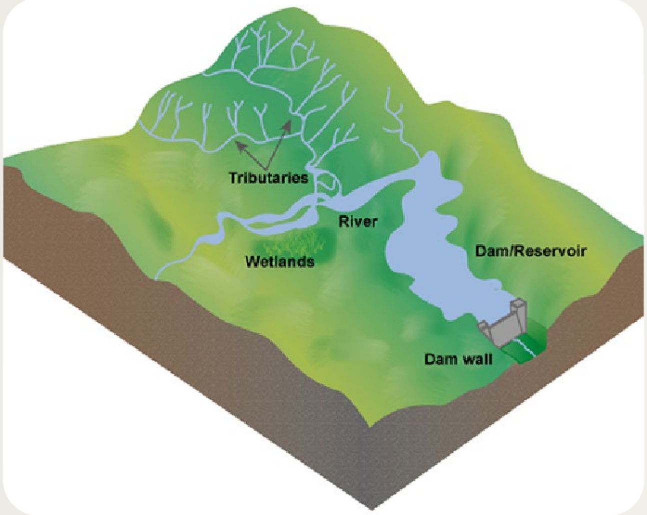
Fresh water use versus available resources

Global warming is adding to the pressure on drinking water supplies and food production all around the world[lvii].

Brief

Design and build solution that can be used by scientists to locate sources of pollution that affect drinking water supplies.

Drinking water comes from a catchment area with tributaries flowing into rivers which in turn flow into reservoirs. Any one of these tributaries could be feeding pollutants into the reservoir and contaminate drinking water. Tracing the source of pollution from a reservoir to the right tributary source enables the water authority to pinpoint the source of solution and rectify it.



Pathways of water pollution, picture source – OpenLearn Create

To trace the source of pollution, scientists need to understand the quality of water in each of the rivers and tributaries feeding a reservoir, and to do this they need a computing solution.

Your brief here is to design a solution, including an app, hardware, and a dashboard.

The solution will enable a scientist to go to a tributary or river and get an instant analysis of whether the water is contaminated or not, without necessarily collecting a sample and sending it back to a lab for analyses. The rationale for this is that in a pollution emergency in a wide water catchment area it may take a long time to get physical water samples back to the lab for analysis, and a mobile solution can enable authorities to get to the source of the problem and deal with it more quickly.

The data collected in the field should then be synchronised with a centralised analysis system that aggregates and visualises data from across the entire catchment area.

There are many ways to measure pollution in water, but its recommended here that the focus should be on electrical conductivity of water – which is based on the principle that the less contaminated the water, the higher its conductivity. In other words, if we know a water sample's conductivity, we have a good indicator of whether its polluted or not.

The app needs to work on a mobile phone, the hardware should be based on an electrical conductivity sensors, micro:bit or Arduino and a PC accessed dashboard should be a web service that takes a data feed from mobile phones.

Phase 1, Define The Problem – 1 hour

Students should first study the problem, then write a clear problem statement –

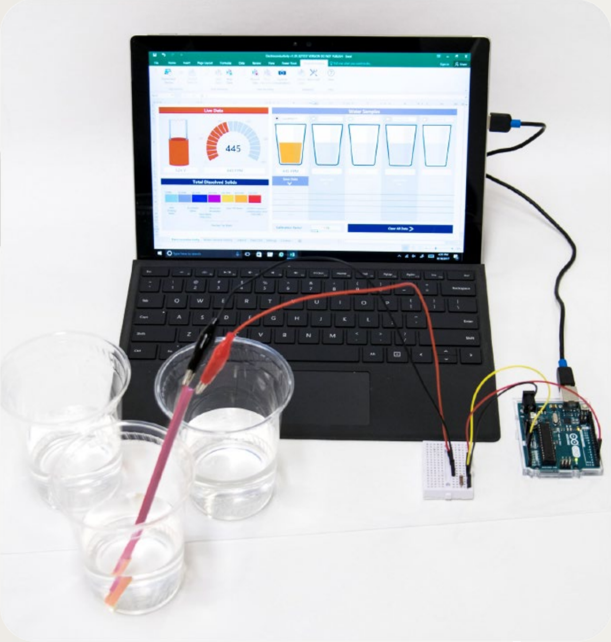
Understand the background to the problem that needs to be solved (20 mins) – https://www.open.edu/openlearncreate/mod/oucontent/view.php?id=80011&printable=1

Understand how water can be tested and analysed using a micro:bit or Arduino with Excel (20 mins) – Measuring Water Quality to Understand Human Impact, H-STEM MWQ, https://learn.microsoft.com/en-us/training/educator-center/instructor-materials/measure-water-quality-understand-human-impact

Write the problem statement (20 mins). You may want to use this as a guide – https://www.startups.com/library/expert-advice/write-best-business-problem-statement

Phase 2, Ideation – 1 hour

Students' ideas should build off their understanding of how to develop a lab-based solution using micro:bit or Arduino with Excel Data Streamer as set out here – H-STEM MWQ, https://learn.microsoft.com/en-us/training/educator-center/instructor-materials/measure-water-quality-understand-human-impact



They should then turn the brief into a set of features for a mobile version of this and develop a flowchart that explains how the solution works.

Think about how the following questions. How can:

- samples be collected and electrical conductivity sensors be used to analyse the samples?
- analysis data be processed by a micro:bit or Arduino?
- data from the micro:bit or Arduino be used in an app on a mobile phone?
- data from the mobile phone be sent to a central aggregation point and visualised on a PC dashboard?
- IoT principles and products be used, if appropriate?

Use this Microsoft Learn module to work out the potential role of IoT in this solution – is it relevant here at all?

Explore Azure IoT services, **ML AIoT**, https://learn.microsoft.com/en-us/training/modules/explore-azure-iot-services/

Students should be careful not to over complicate their solution design. The basic requirements are

Test water sample > get data from test to phone > synchronise data from the phone to the cloud > aggregate and visualise the data so decisions can be made both locally and centrally.

Phase 3, Build – 3 hours

Technology Focus

Develop a working prototype of the solution focussing on the following types of technologies –

Windows Azure Mobile Services for Android or iOS:

- Android – **ML WAMSfA**, https://learn.microsoft.com/en-us/shows/windows-azure-mobile-services/android-getting-started-data-connecting-your-app-to-windows-azure-mobile-services
- iOS – **ML WAMSfiOS**, https://learn.microsoft.com/en-us/shows/windows-azure-mobile-services/ios-getting-started-data-connecting-your-app-to-windows-azure-mobile-services

Power BI – https://powerbi.microsoft.com/en-gb/

Microsoft Learn Modules

Use the following Microsoft Learn Modules to help develop the solution –

ML4S WAVS, Publish a web app to Azure with Visual Studio, https://docs.microsoft.com/en-us/learn/modules/publish-azure-web-app-with-visual-studio/

ML CAAS, Configure Azure App Services, https://learn.microsoft.com/en-us/training/modules/configure-azure-app-services/

Phase 4, Present and Evaluate – 1 hour 40 mins

Assessment of this project is via group evaluation of projects, and teacher assessment of contributions to the team by individual members.

Each team, ideally 5 diverse members of mixed ability, is given 5 mins to present their solution in slides and a demo, followed by 5 min Q&A. All other students in the class are encouraged to ask questions and one way to do this is to award team points for good questions.

Ideally, each team will make a video of their solution and presentation for future reference.

Supporting Content

- https://www.open.edu/openlearncreate/mod/oucontent/view.php?id=80011
- w3schools SQL: https://www.w3schools.com/sql/
- https://www.khanacademy.org/computing/computer-programming/sql
- https://www.w3schools.com/js/default.asp
- https://www.w3schools.com/nodejs/default.asp

Coding Certification Option

Students can gain certified recognition for their coding skills though "Coding in Minecraft" – a computer science credential program with supporting curriculum, delivered through Minecraft Education.

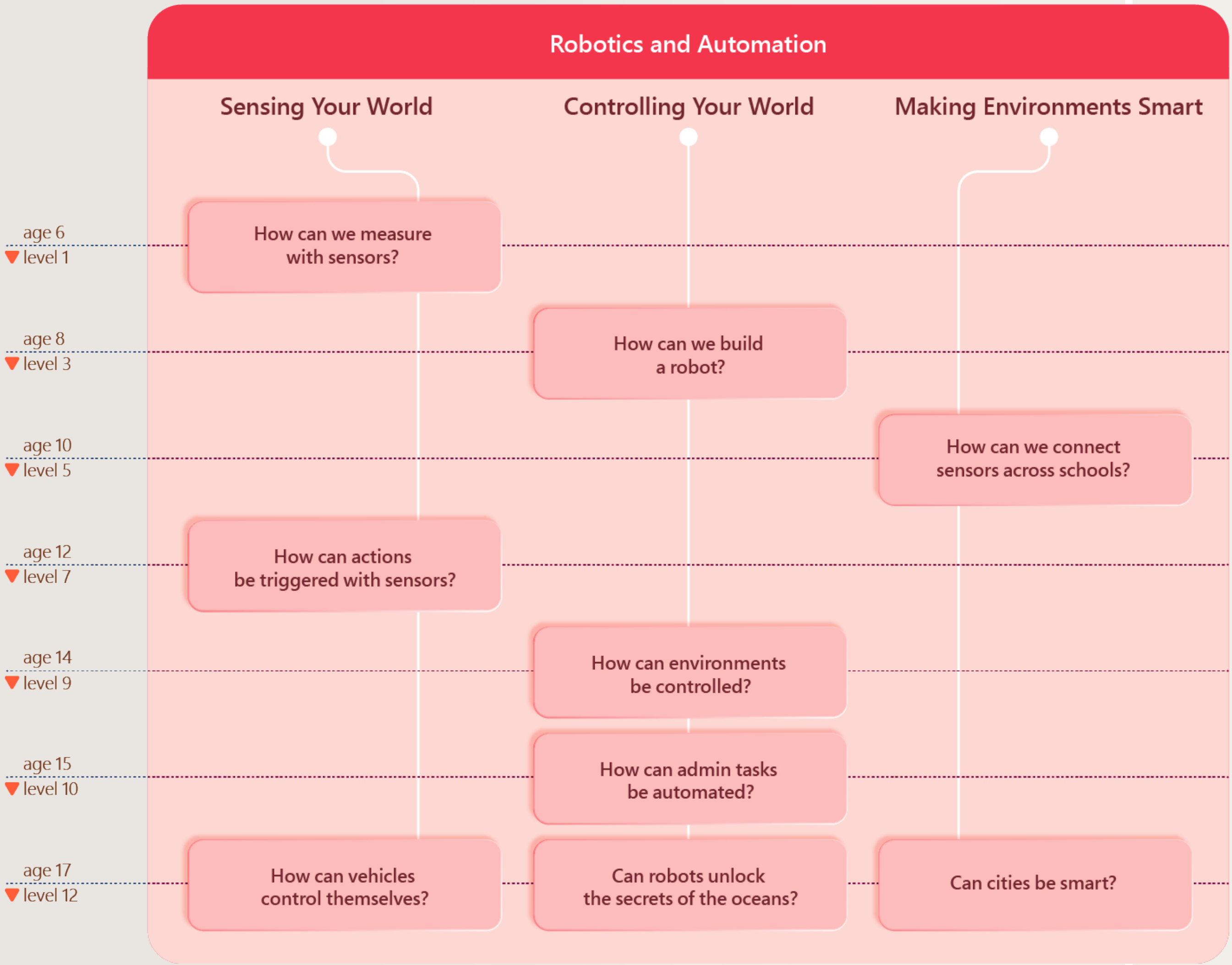
The pathway comprises four courses that immerse students in a Minecraft world to develop and prove their coding skills using MakeCode and JavaScript or Python. Find out more here https://codingcredentials.com

Robotics and Automation

The world of robotics is where CS, applied sciences, engineering and mathematics come together to offer a rich space for learning. Whilst there are many school programmes focussing on physical robotics, there is very little concentrating on the automation of cognitive tasks. Yet cognitive task automation through Robotic Process Automation (RPA) is a major and growing[lviii] disruptive force across the world of work.

The goal of this Domain is to develop both the physical robotic and RPA skills by using three key Big Ideas:

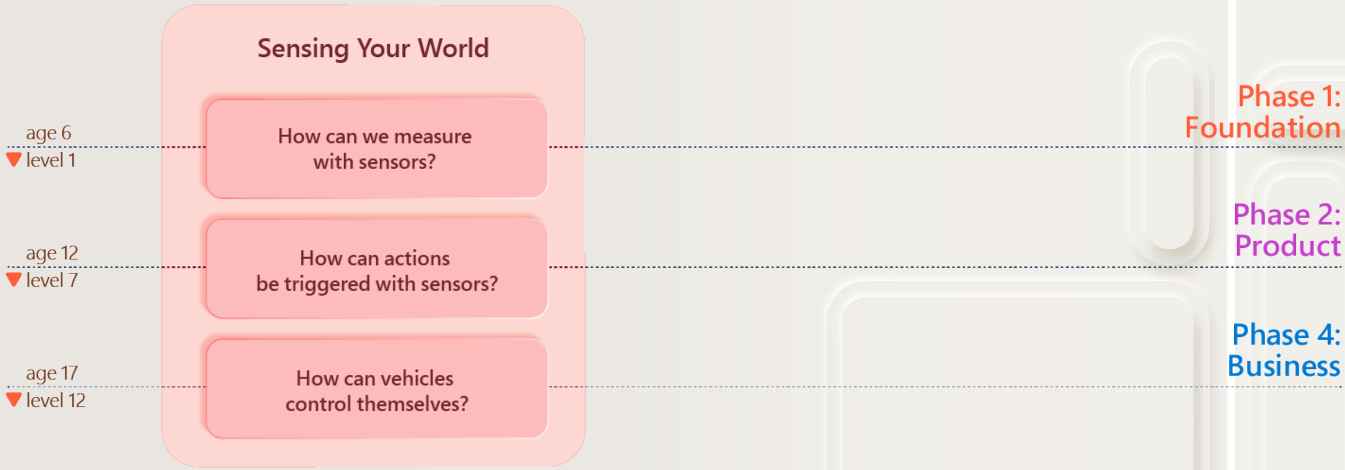
- Sensing Your World
- Controlling Your World
- Making Environments Smart



Robotics and Automation

	Sensing Your World	Controlling Your World	Making Environments Smart
age 6 ▼ level 1	How can we measure with sensors?		
age 8 ▼ level 3		How can we build a robot?	
age 10 ▼ level 5			How can we connect sensors across schools?
age 12 ▼ level 7	How can actions be triggered with sensors?		
age 14 ▼ level 9		How can environments be controlled?	
age 15 ▼ level 10		How can admin tasks be automated?	
age 17 ▼ level 12	How can vehicles control themselves?	Can robots unlock the secrets of the oceans?	Can cities be smart?

Sensing Your World



	Sensing Your World	
age 6 ▼ level 1	How can we measure with sensors?	Phase 1: Foundation
age 12 ▼ level 7	How can actions be triggered with sensors?	Phase 2: Product
age 17 ▼ level 12	How can vehicles control themselves?	Phase 4: Business

Understanding robotics and automation is predicated on understanding how data can be gathered from sensors and then processed.

At the start of this learning path is understanding how probes can be used to gather data, and how that data can then be visualised. Further along this journey students will learn how autonomous vehicles use sensors extensively and will be introduced to the concept of artificial intelligence. As learners progress the focus broadens to sensing a wide range of physical phenomenon including gases – laying the foundations for the 'Making Environments Smart' Big Idea.

How can we measure with sensors?

Domain – Robotics

Idea – Sensing Your World

Level – 1

Time allocation – 10 hours

Learning Goals

Main

- Explain that computers collect data from various input devices, including sensors and application software, CAS H&P 3
- Collect, visualise and explain patterns in data, MCSG
- Recognise that data can be structured in tables to make it useful, CAS D&DR 1

Supplementary

- Understand that sensors display values
- Learn how to plug a probe into a computer; measure a physical phenomenon; read, watch and make sense of the numbers changing
- Predict how numbers will change when sensed conditions change
- Appreciate that data is generated from sensors, and can be visualised
- Know that temperature, light and sound can be 'measured'

Concepts

- Sensors help us understand and control our world
- Sensors use electronics and computers to gather data
- Data from sensors gives us useful information and insights
- Conductivity and the flow electricity in a current

Challenges

Computational Thinking

- Set up a basic sensing experiment, e.g. using a temperature probe
- Make a fruit keyboard using micro:bit
- Use sensors in a Little Bits circuit to trigger an action with a sensor

Data Literacy – Appreciate the basics of time series, for example, the change of temperature over time visualised in a spreadsheet

Design Thinking – Produce and compare bar charts and line graphs

Supporting Content

Microsoft content

- **MC USB**, WebUSB, explore sensors on Micro:Bit, https://makecode.microbit.org Deep Dive
- **MC M**, Music, including a 'banana keyboard', https://makecode.microbit.org Music

Supplementary content

- Makey Makey lesson 1 to 8 – https://makeymakey.com/pages/how-to
- Getting Started with Little Bits – https://classroom.littlebits.com/kits/educator-quick-start-guide
- Primary Science Curriculum Activities with Sensors, Cambridge University – http://oer.educ.cam.ac.uk/wiki/Primary_Science_Curriculum_Activities_with_Sensors
- Sensors and Graphs in Primary Schools, University of Amsterdam – https://www.iederkindeentalent.nl/wp-content/uploads/2012/06/sciencesensors1.pdf

How can actions be triggered with sensors?

Domain – Robotics

Idea – Sensing Your World

Level – 7

Time allocation – 10 hours

Learning Goals

Main

- Use C code functions, variables and structures in a control solution, MCSG
- Apply practical experience of a high-level textual language, CAS P&D 5
- Process Serial Data in a control solution, MCSG

Supplementary

- Develop and use Arduino skills
- Learn how an ultrasonic sensor works and how to use one in a robot

Concepts

- Sensors help us understand and control our world
- Sensors use electronics and computers to gather data
- Data from sensors gives us useful information and insights
- Conductivity and the flow electricity in a current

Challenges

Computational Thinking – build a floorbot

Data Literacy – analyse, use and manipulate serial data from the ultrasonic sensor to trigger motor actions

Design Thinking – design the floorbot and its functions

Supporting Content

Microsoft content

- **MC Tys**, Toys, https://makecode.microbit.org Toys
- **MC Sci**, https://makecode.microbit.org Science
- **MS DS**, Data-Streamer, https://support.microsoft.com/en-us/topic/data-streamer-c90aebcf-3d44-47ab-a068-549a0b9edfc6?ui=en-us&rs=en-us&ad=us
- **H-STEM HDO**, How deep is the ocean? Modelling and graphing the ocean floor with an ultrasonic sensor, https://education.microsoft.com/pt-pt/lesson/4e1b5c7f"
- **H-STEM HFHR**, How fast do humans run? https://www.microsoft.com/en-us/education/education-workshop/comparing-speeds.aspx

Supplementary content

- Obstacle-Avoiding Robot Using Arduino with Ultrasonic Sensor Develop Arduino programs with Microsoft Visual Studio – https://create.arduino.cc/projecthub/chandankumarcmsn/obstacle-avoiding-robot-using-arduino-with-ultrasonic-sensor-af66f1

How can vehicles control themselves?

Domain – Robotics and Automation

Idea – Sensing Your World

Level – 12

SDG – 9. Industry, Innovation and Infrastructure

Time allocation – 6:40

Learning Goals

Main

- Describe how artificial intelligence drives many software and physical systems, CSTA 3B-AP-08
- Integrate different types of sensors in an autonomous system, MCSG

Supplementary

- Apply a range of mechatronic skills to solve robotics problems
- Understand that robots comprise mechanical, sensory and control systems
- Know how robots can be used to automate physical processes in the real world

Concepts

- An autonomous vehicle is one that is able to operate itself without any human intervention, through the ability to sense its surroundings
- We grade autonomy with five levels from 1: 'Driver Assistance' up to 5: 'Full Automation'
- AI is central to the control function of autonomous vehicles

Challenges

Computational Thinking – Program an embedded computer in a model autonomous vehicle

Data Literacy – Analyse data from a model autonomous vehicle

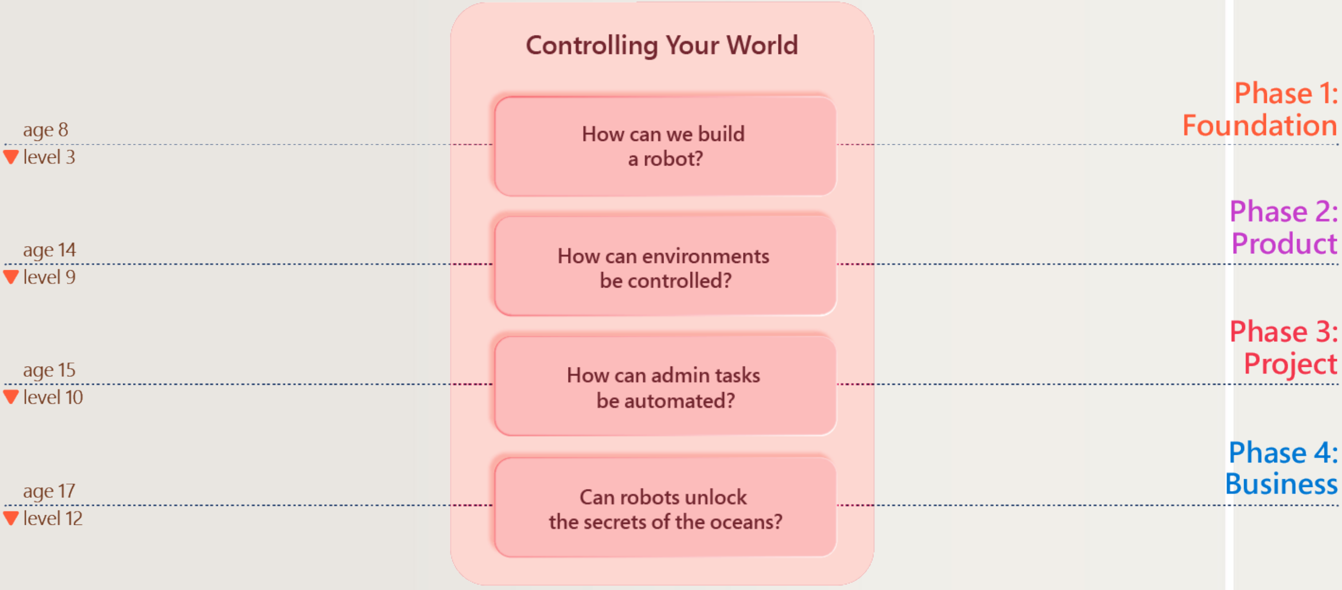
Design Thinking – Build high-level blueprints for an autonomous vehicle control system

Supporting Content

Microsoft content

- AiAD, Azure in Autonomous Driving, https://autonomous-driving.org/wp-content/uploads/2018/10/Microsoft_Azure-in-Autonomous-Driving.pdf

Controlling Your World



This Big Idea focuses on building robots. At the most basic level students will learn how to programme simple toy-like robots. They will then progress to using data from sensors and programming embedded computers to control physical processes.

Further along this learning journey, students will be introduced to the concepts of 3DP, Internet of Things (IoT) and RPA. They will be set challenges on authentic problems linked to helping people with disabilities and the UN's SDGs.

How can we build a robot?

Domain – Robotics

Idea – Controlling Your World

Level – 3

Time allocation – 10 hours

Learning Goals

Main

- Program robots, MCSG
- Identify your position and calculate the position of a landmark or object relative to your position in the real-world, MCSG

Supplementary

- Understand block programming
- Understand how robots use data and algorithms to control movement and execute tasks

- Understand that robots comprise mechanical, sensory and control systems
- Know how robots can be used to automate physical processes in the real world

Concepts

- Robots are coded
- Changes in data can be used to trigger events
- Robots can be used to automate tasks that humans may otherwise have done

Challenges

Computational Thinking

- Use a games environment to control events with code
- Use block-based programming to control a robot

Data Literacy

- Use cartesian coordinates with a robot
- Analyse media reports about the impacts of robots on humanity, trends and predictions

Design Thinking – Design and build a robot toy

Supporting Content

Microsoft content

- MC Cue, Cue Robotic system, https://www.makewonder.com/robots/cue/explore/
- MEE PRR, Patterns, Repetition, Rhythm, https://education.minecraft.net/en-us/challenges/patterns-repetition-rhythm
- MC GRG, Games and Radio Games, https://makecode.microbit.org Games, Radio Games
- MC Tls, Tools, https://makecode.microbit.org Tools
- MC MVP, Making a Virtual Pet, https://www.youtube.com/watch?v=zc3njwmP_O8&list=PLmqeu38gRdJVCMUhgmF8OrjOhYpYtoh9U
- MC PW, Plant watering, https://makecode.microbit.org/projects/plant-watering

Supplementary content

- Using robots to teach mathematics https://www.cmu.edu/roboticsacademy/PDFs/Research/SilkSchunn2008a-ASEE-Presentation.pdf
- Computer Science Unplugged (CSU 8), Kidbots – https://csunplugged.org/en/topics/kidbots/

How can environments be controlled?

Domain – Robotics and Automation

Idea – Controlling Your World

Level – 9

Time allocation – 10 hours

Learning Goals

Main

- Apply combined control technologies in a Cloud-based IoT solution, MCSG
- Develop, apply, interpret and communicate statistical models, judgements and arguments, MCSG

Supplementary

- Understand the basics of Internet of Things (IoT)
- Know how sensors, data, algorithms and devices can work together
- Use threshold data trigger alerts and automated responses
- Use data from sensors and big data sets for data driven decision-making
- Naturally interact with agents and models to get insights

Concepts

- Internet of Things
- Machine Learning
- Smart environments
- Smart agriculture

Challenges

Computational Thinking – Work with big data, train models, define events and triggers

Data Literacy – Gather and analyse data, set and use thresholds

Design Thinking – Design a smart farm

Supporting Content

Microsoft content

- **H-STEM BMEH**, Building Machines That Emulate Humans. Robotic models to understand the biomechanics of the human hand, https://www.microsoft.com/en-us/education/education-workshop/robotic-hand.aspx
- **MC A-Z R**, A-Z Robotics, https://tinkerspark.teachable.com
- **MC CJ RR**, CodeJoy Remote Robotics, https://codejoyeducation.com
- **MC F**, Fashion, https://makecode.microbit.org Fashion
- **MC LC**, Live Coding, https://makecode.microbit.org Live Coding
- **H-STEM FbIoT**, FarmBeats for Students, https://education.microsoft.com/en-us/lesson/5d991297

How can admin tasks be automated?

Domain – Robots and Automation

Big Idea – Controlling your World

Level – 10

SDG – 8. Decent Work and Economic Growth

Time allocation – 10 hours

Learning Goals

Main

Program an RPA process, MCSG

Supplementary

- Demonstrate the ability to automate administrative tasks on a server using a scripting language such as Bash
- Choose and implement appropriate RPA for manual UI tasks

- Distinguish the differing solutions for automation and their appropriate use cases

Concepts

- Robotic Process Automation (RPA) handles repetitive business processes by mimicking human actions in common UIs without the need for systems integration
- Bash scripting allows the user access to administer files, users and functionality on a server
- RPA may be enhanced with AI to create RPA processes that can learn

Challenges

Computational Thinking – Write a Bash script to check user activity on a server

Data Literacy – Summarize and interpret the output from an administrative Bash script

Design Thinking – Choose an appropriate UI workflow for attended/unattended automation using RPA

Supporting Content

Microsoft content

- **ML4S QnA**, Build a bot with QnA Maker and Azure Bot Service, https://docs.microsoft.com/en-us/learn/modules/build-faq-chatbot-qna-maker-azure-bot-service/ Learn MPPF,
- **Ibot**, Introduction to Bots, https://www.microsoft.com/en-us/ai/ai-school
- **LRN MPPF**, Microsoft Power Platform Fundamentals, https://docs.microsoft.com/en-us/learn/paths/power-plat-fundamentals/
- **LRN CBPVA**, Create bots with Power Virtual Agents, https://docs.microsoft.com/en-us/learn/paths/work-power-virtual-agents/
- **ML4S I2B**, Introduction to Bash, https://docs.microsoft.com/en-us/learn/modules/bash-introduction/

Can robots unlock the secrets of the oceans?

Domain – Robotics and Automation

Idea – Controlling Your World

Level – 12

SDG – 14. Life Below Water

Time allocation – 6:40 hours

Learning Goals

Main

Apply a range of mechatronic skills to solve robotics problems, MCSG

Supplementary

- Demonstrate understanding of the applications of mechatronics for science and industry
- Demonstrate mastery of the basic principles of mechatronics such as sensors, activators and control systems

Concepts

- Mechatronics combines electrical, ICT and mechanical engineering to build highly functional robotics
- Mechatronics has broad applications in science and industry ranging from smart elevators to planetary exploration rovers

- A servomechanism (servo) such as an antilock brake uses error-sensing negative feedback to correct its action

Challenges

Computational Thinking – Program a robot on the Moveit! Platform to recognize, pick up and hold objects

Data Literacy – Analyse spatial feedback from a robot to determine its position and trajectory

Design Thinking – Design a robot to scour the ocean floor for samples using a 3-D design tool such as Gazebo

Supporting Content

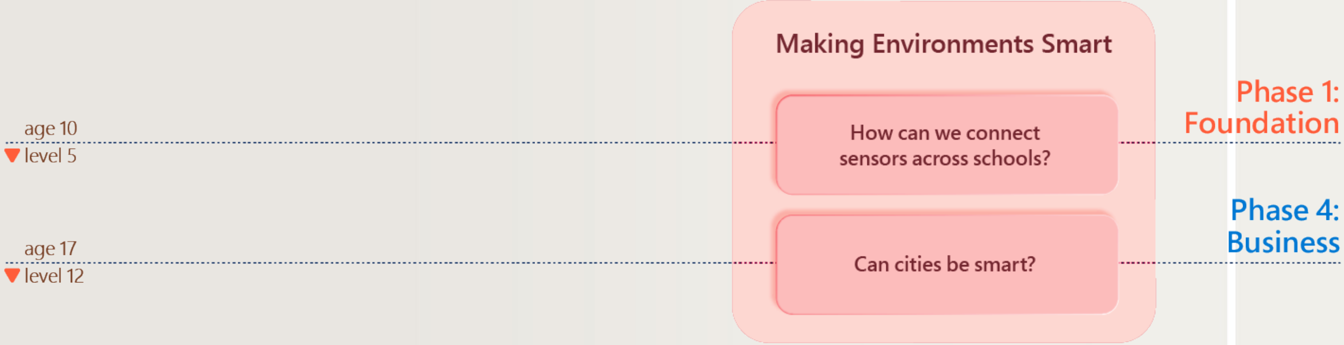
Microsoft content

- MS AS, Autonomous Systems, https://docs.microsoft.com/en-us/autonomous-systems/

Supplementary content

- https://www.bbc.co.uk/news/av/technology-48747238/how-robo-boat-tech-could-help-uncover-ocean-secrets

Making Environments Smart



Having built foundations of understanding in sensors and robotics students can use this Big Idea to focus on how to use robotics at-scale in smart cities and smart environments. Key focal points of learning will be to push deeper into Internet of Things, understand open data, and learn about the use of computing to deliver public services.

The authentic problem-based learning context for this Big Idea involves using IoT to address global problems identified in the SDG such as hunger, clean energy and climate action.

How can we connect sensors across schools?

Domain – Robotics and Automation

Idea – Making Environments Smart

Level – 5

Time allocation – 10 hours

Learning Goals

Main

Explain the basic idea of how IoT can be combined with Cloud Computing to aggregate and visualise data from around the world, MCSG

Supplementary

- Explain the basics of how sensors can be used to gather environmental data
- Explain the basics of how data can be visualised and used to make decisions

Concepts

- The Cloud
- International collaboration online
- The more data you have, the better insights you will get

Challenges

Computational Thinking – Measure, upload, explore, and share weather data between your school and schools in other countries

Data Literacy – work across a wide range of metrological units of measurement and build data stories

Design Thinking – design weather data experiments and analysis, and visualisations

Supporting Content

Microsoft content

- **H-STEM AWA**, Analyzing Windspeed With Anemometers, Excel, Arduino, https://www.microsoft.com/en-us/education/education-workshop/anemometer.aspx
- **MC DD**, Deep Dive, https://makecode.microbit.org Deep Dive
- **MC BMCH**, Behind the MakeCode Hardware, https://makecode.microbit.org Behind the MakeCode Hardware

Supplementary content

- IoT@School – https://exploratory.sciencescope.uk
- IoT@School activities – https://exploratory.sciencescope.uk/category/exciting-ways-to-use-iotatschool-in-the-classroom/
- https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/eco-calculator

Can cities be smart?

Domain – Robotics and Automation

Idea – Making Enviroments Smarter

Level – 12

SDG – 11. Sustainable Cities and Communities

Time allocation – 6:40 hours

Learning Goals

Main

Apply a range of IoT skills to solve smart city/smart environment problems, MCSG

Supplementary

- Demonstrate an understanding of the economic, social and governmental applications of IoT technology
- Demonstrate an understanding of potential evolution of IoT in the Fourth Industrial Revolution

Concepts

- A Smart City makes intelligent use of information flows, including from IoT devices, to guide civic governance at a strategic, tactical and operational level
- A Smart Grid makes intelligent use of the data gathered from elements of the grid to distribute and manage energy in order to meet efficiency, renewability and other goals
- The Industrial Internet of Things (IIOT) is the use of IoT in industrial sectors and applications including medicine and manufacturing
- The Fourth Industrial Revolution is the theory that digital technologies will fuse with the physical and biological spheres to unleash a wave of innovation bringing sustainable economic growth

Challenges

Computational Thinking – Write a program to gather data from a sensor device and process it into useful information

Data Literacy – Map data structures from raw sensor input into tabular form suitable for analytics

Design Thinking – Diagram the components, data flows and software architecture of an IoT system that visualizes environmental conditions in real time

Supporting Content

Microsoft content

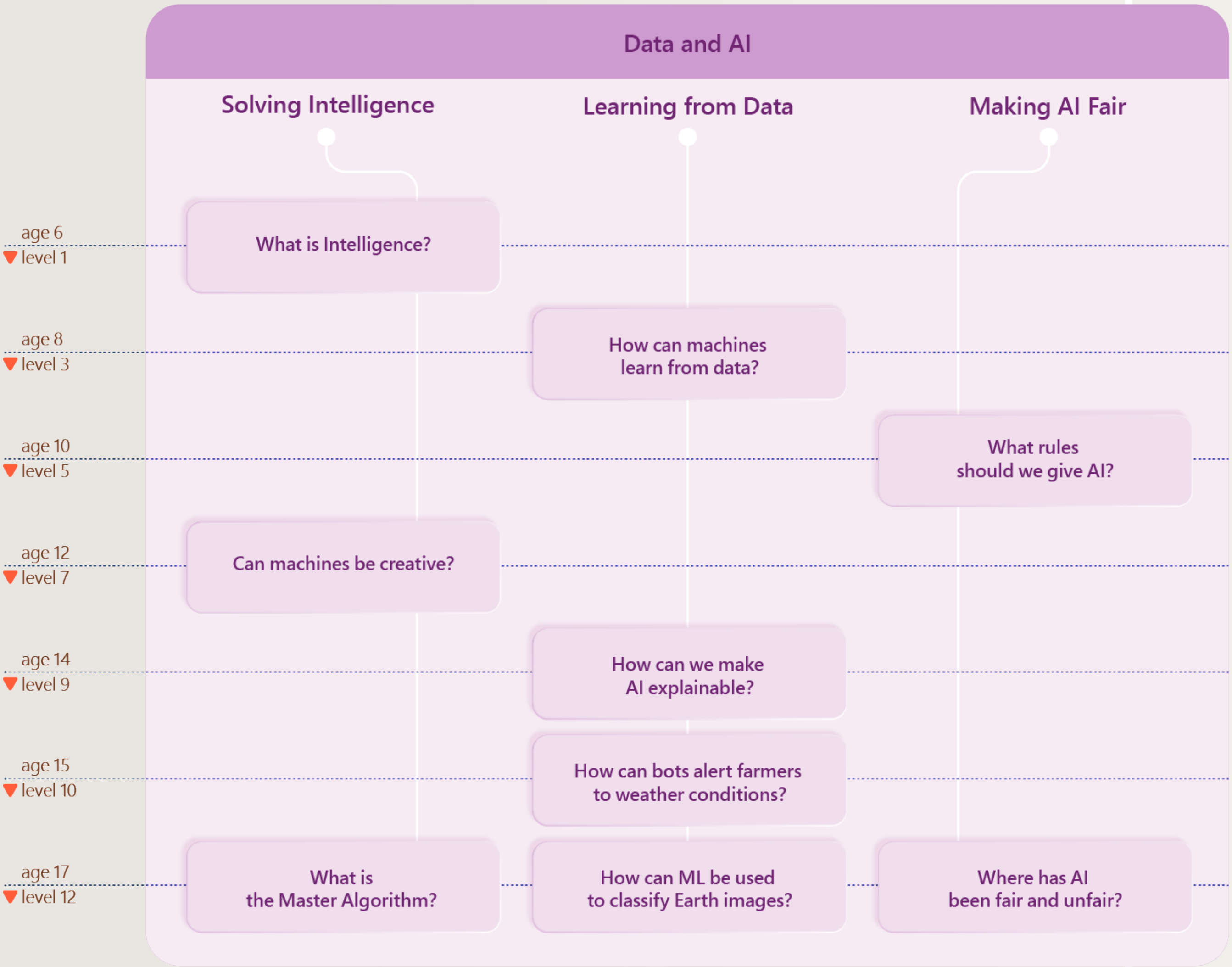
- **SCS**, Smart City Solutions, https://clouddamcdnprodep.azureedge.net/gdc/gdcSKdGhr/original
- **ML4S AIoT**, Choose the best Azure IoT service for your application, https://docs.microsoft.com/en-us/learn/modules/iot-fundamentals/
- **AIoTDK**, Azure IoT Developer Kit, https://microsoft.github.io/azure-iot-developer-kit/

Data and AI

There is wide consensus that, as the Royal Society (UK) put it in 2018 “AI will affect occupations at all levels of pay and education”, and “all occupations could be transformed by AI to some extent”.[lix] This means that children need to be adequately prepared for working with, and using AI.

The goal of this Domain is to develop skills in building basic AI systems by using the following Big Ideas:

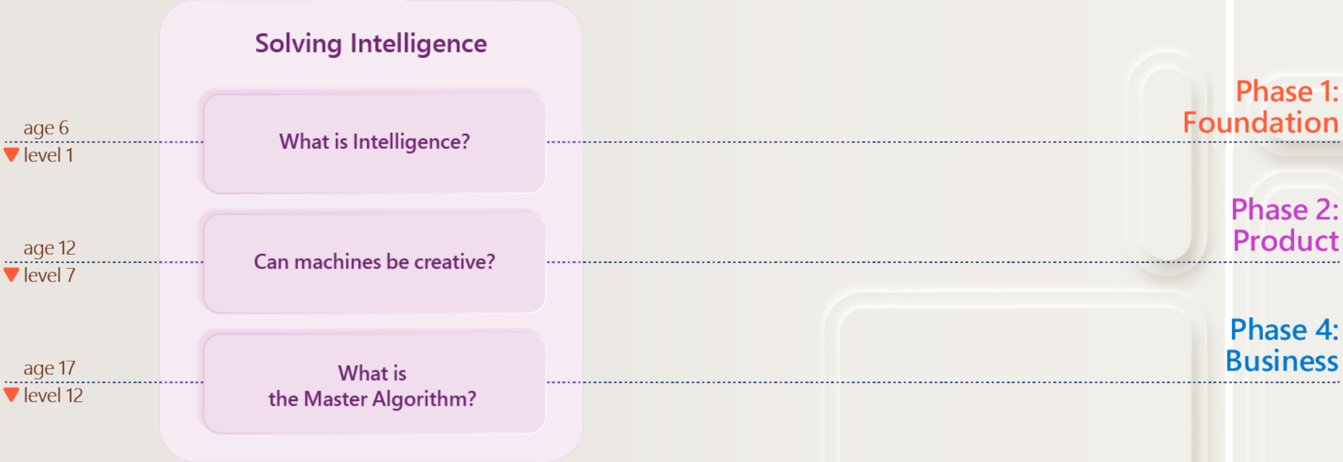
- Solving Intelligence
- Making AI Fair
- Learning from Data



Data and AI

	Solving Intelligence	Learning from Data	Making AI Fair
age 6 ▼ level 1	What is Intelligence?		
age 8 ▼ level 3		How can machines learn from data?	
age 10 ▼ level 5			What rules should we give AI?
age 12 ▼ level 7	Can machines be creative?		
age 14 ▼ level 9		How can we make AI explainable?	
age 15 ▼ level 10		How can bots alert farmers to weather conditions?	
age 17 ▼ level 12	What is the Master Algorithm?	How can ML be used to classify Earth images?	Where has AI been fair and unfair?

Solving Intelligence



	Solving Intelligence	
age 6 ▼ level 1	What is Intelligence?	Phase 1: Foundation
age 12 ▼ level 7	Can machines be creative?	Phase 2: Product
age 17 ▼ level 12	What is the Master Algorithm?	Phase 4: Business

This Big Idea is about understanding the quest to emulate and surpass human intelligence with machines. Key concepts include understanding the meaning of the words 'intelligence' and metacognition, and the distinctions between mind and brain. Building on this understanding of intelligence itself, learners can begin to make sense of machine intelligence.

As they progress in this Big Idea, learners will focus on areas such as Artificial General Intelligence, or The Singularity – and questions such as “can machines be creative?” Most importantly they will understand what AI is and what it is not, and the possibilities and limits of AI.

What is Intelligence??

Domain – Data and AI

Big Idea – Solving Intelligence

Level – 1

Time allocation – 10 hours

Learning Goals

Main

Understand that computers have no intelligence and that computers can do nothing unless a program is executed, CAS H&P

Supplementary

- Develop a baseline understanding of the meaning of the word 'intelligence'
- Help learners to take control of their learning
- Develop a baseline understanding of machine intelligence

Concepts

- Metacognition
- Mind vs brain
- Can machines have thoughts?

Challenges

Computational Thinking

- Stack rank objects and living things by how intelligent they are
- Play a ‘human neural network’ game
- Work out if a given chatbot is a human or a machine by asking it questions

Data Literacy – Understand how simple algorithms can be chained together

Design Thinking – Visually represent different types of thinking processes, comparing the things that computers can do

Supporting Content

Microsoft content

- **MEECodWM 9**, AI, explore the concept of artificial intelligence, https://education.minecraft.net/en-us/lessons/unit-9-ai
- **ExpAI**, Experience AI, https://www.microsoft.com/en-us/ai/experience-ai

Supplementary content

- “How to Make a Mind”
- Can AI Guess Your Emotion? https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/ai-lesson-plans/can-ai-guess-your-emotion

Can machines be creative?

Domain – AI

Big Idea – Solving Intelligence

Level – 7

Time allocation – 10 hours

Learning Goals

Main

Understand how data can be used in the creative world, MCSG

Supplementary

- Ask questions about the nature of creativity
- Understand that AI has limits

Concepts

- There are limits to what AI can do
- The Imitation Game
- Divergent vs convergent thinking

Challenges

Computational Thinking – Explain the roles of decomposition, abstraction, pattern recognition and algorithms in computation

Data Literacy – Use data to create art

Design Thinking – Design images, written pieces, or sound using algorithms

Supporting Content

Microsoft Content

- **ICJ 1**, Fundamentals of Artificial Intelligence, https://imaginecup.microsoft.com/en-us/category/14?skillLevel=0
- **ML4S I2T**, Introduction to Translator, https://docs.microsoft.com/en-us/learn/modules/intro-to-translator/

Supplementary Content

- "AI: More Than Human", Barbican virtual exhibition – https://artsandculture.google.com/project/ai-more-than-human
- The Creativity Code, Art and Innovation in the Age of AI, Marcus du Sautoy
- 41 Creative Tools to Generate AI Art, https://aiartists.org/ai-generated-art-tools

What is the Master Algorithm?

Domain – AI

Big Idea – Solving Intelligence

Level – 12

SDG – 8. Decent Work and Economic Growth

Time allocation – 6:40 hours

Learning Goals

Main

- Discuss the "five tribes of AI" MCSG,
- Demonstrate understanding of the principles of Deep Learning and its power, potential and limitations, MCSG

Supplementary

- Discuss and assess the power and applicability of various AI approaches to practical problems
- Critically analyse the "instrumental goals" of Superintelligence
- Compare the main ideas behind the Master Algorithm and Superintelligence

Concepts

- There are five tribes of AI, each with a different approach: symbolists, connectionists, evolutionaries, Bayesians and apologizers.
- There is no one correct way to develop AI, but several competing 'vectors'
- Deep Learning is the latest 'master algorithm' but is it a basis for Artificial General Intelligence?
- Is it possible to have a perfect understanding of how the world works and how people work within the world?

Challenges

Computational Thinking – propose experiments that illustrate the Big Ideas of each of the “five tribes of AI"

Data Literacy – Describe the flow of data through a deep learning network for classification problems

Design Thinking – Design ways to illustrate the main Big Ideas behind the Master Algorithm and Superintelligence

Supporting Content

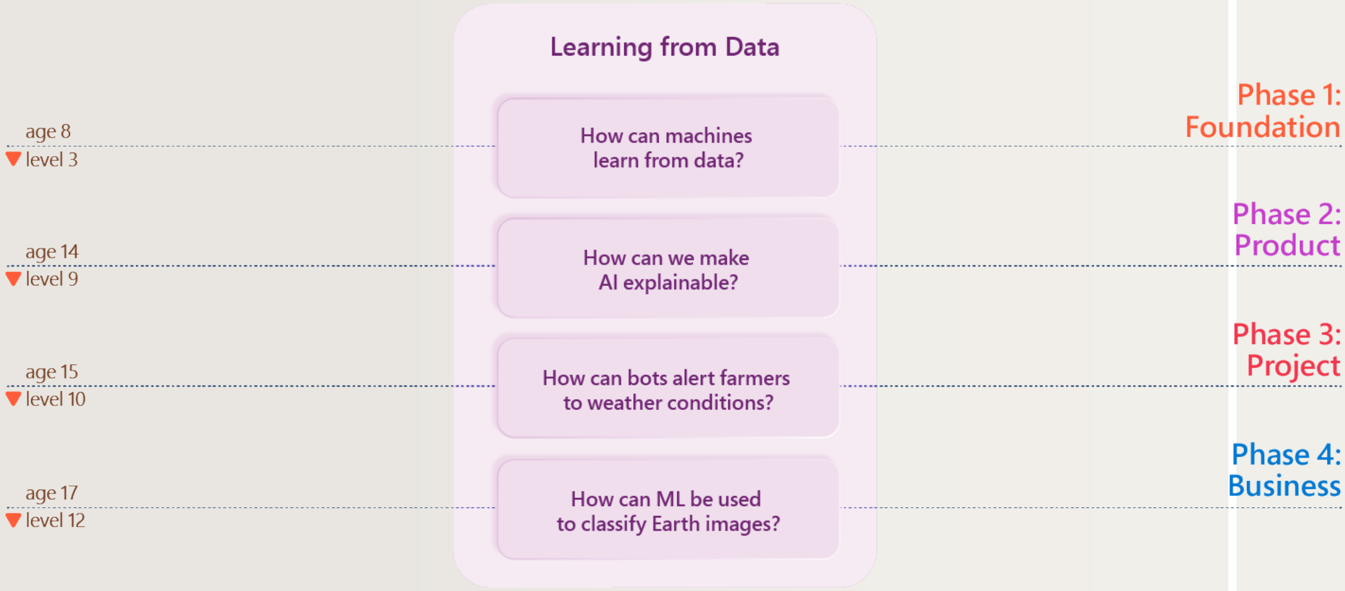
Microsoft content

- **Int2AI**, Introduction to AI, https://www.microsoft.com/en-us/ai/ai-school
- **DLvML**, Deep Learning Versus Machine Learning, https://docs.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs-machine-learning

Supplementary content

- Recommended by Bill Gates as the two books everyone should read to understand AI
    - '*The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our* World', by Pedro Domingos
    - '*Superintelligence*' by Nick Bostrom
- https://www.coursera.org/learn/intro-to-deep-learning
- https://www.datacamp.com

Learning from Data


Learning from Data
age 8
level 3
How can machines learn from data?
Phase 1: Foundation
age 14
level 9
How can we make AI explainable?
Phase 2: Product
age 15
level 10
How can bots alert farmers to weather conditions?
Phase 3: Project
age 17
level 12
How can ML be used to classify Earth images?
Phase 4: Business

This Big Idea focusses on learning how to build machine learning solutions. At an entry level, the MCSG provides unplugged activities that enable key concepts such as reward-based learning to be learned in a fun way. This Big Idea offers direct connections to students' everyday lives through activities such as examining the data generated on mobile phones and understanding how it is processed by a company that employs artificial intelligence.

Further along the learning pathway, students learn how to write simple machine learning experiments to uncover insights from messy data.

Equipped with an understanding of what AI can and cannot do, an attitude of making sure that AI is fair and the ability to produce basic machine learning solutions, students can then turn their attention to authentic problem-based learning projects based on SDG goals.

How can machines learn from data?

Domain – Data and AI

Big Idea – Learning From Data

Level – 3

Time allocation – 10 hours

Learning Goals

Main

- Express probabilities numerically, *MCSG*
- Understand that data can be learned from, *MCSG*

Supplementary

- Develop an understanding of the idea of machines being able to 'learn'
- Appreciate that controlling data and bias is everyone's responsibility

Concepts

- When you use your phone, you are contributing to AI systems

- AI protagonism – control AI or have it done to you
- It's important to have rules to control the use of advanced technology

Challenges

Computational Thinking – "Sweet Computer" board game, Chatbot, Turing Test

Data Literacy – Play and analyse probability games such as Tic Tac Toe, Rock Paper Scissors, coin tosses and dice throws, chatbots that learn

Design Thinking – How to Make a Mind

Supporting Content

Microsoft content

- **H-STEM NC**, NASA Collection, including Analyzing the astronauts' photos of Earth to predict climate change, https://education.microsoft.com/en-us/hackingStem/nasa
- **MDL WWC 8**, Working With Computers, Getting Started With Excel, https://www.linkedin.com/learning/working-with-computers-and-devices/working-with-computers-to-create-opportunity?autoAdvance=true&autoSkip=false&autoplay=true&resume=true
- **H-STEM DoDN** Pt 1, Day of Data: NASA Rockets, Stage 1 – Introduction to Data, https://education.microsoft.com/en-us/resource/c4db6ddd
- **H-STEM DoDN** Pt 2, Day of Data: NASA Rockets, Stage 2 – Introduction to Data Visualization, https://education.microsoft.com/en-us/resource/c4db6ddd
- **MC CSP AP 6**, Computer Science Principles course, Unit 6 – Data Science, https://microsoft.github.io/makecode-csp/unit-6/data-science
- **MEE I2AIC** 1-5, Introduction to AI Concepts, units 1-5, pattern recognition, machine learning, use of live data, eg https://education.minecraft.net/en-us/lessons/ai-1-who-is-that-ocelot

Supplementary content

- **CS4FN** – Machine Learning Board Game, http://www.cs4fn.org/machinelearning/sweetlearningcomputer.php
- Computer Science Unplugged (CSU 10), The Turing Test – https://classic.csunplugged.org/the-turing-test/
- Introduction to Probability https://www.khanacademy.org/math/probability/probability-geometry/probability-basics/a/probability-the-basics
- CS Field Guide, NZ, 2 Artificial Intelligence https://csfieldguide.org.nz/en/chapters/artificial-intelligence/

How can we make AI explainable?

Domain – AI

Big Idea – Making AI Fair

Level – 9

Time allocation – 10 hours

Learning Goals

Main

- Explain how data is encoded from a range of media and used to make predictions, MCSG
- Explain the effects of bias, MCSG
- Discuss issues of bias and accessibility in the design of existing technologies, CSTA 2-IC-21
- Query data on one table using a typical query language, CAS D&DR 5

Supplementary

- Know why it is important to understand why AI makes the predictions they do
- Understand the importance of simplification in Computer Science
- Understand the key tenets of Explainable AI (XAI)

Concepts

- AI is complex but needs to be widely understood for ethical, societal, safety and democratic reasons
- XAI is a methodology with tools that is used to explain artificial intelligence to non-experts
- It is increasingly important for both the producers and consumers of AI to understand it

Challenges

Computational Thinking – Explain the roles of decomposition, abstraction, pattern recognition and algorithms in computation

Data Literacy – Query messy data in a table, and find bias

Design Thinking – Explain a simple Machine Learning model – e.g. KNN in a spreadsheet

Supporting Content

Microsoft content

- **ML4S FDS**, Foundations of data science, https://docs.microsoft.com/en-us/learn/paths/foundations-data-science/
- **ML4S IDML**, Introduction to data for machine learning, https://docs.microsoft.com/en-us/learn/modules/introduction-to-data-for-machine-learning/
- **ML4S DCM**, Data collection and manipulation, https://docs.microsoft.com/en-us/learn/modules/collect-manipulate-data-python-nasa/
- **ML4S UCM**, Create and understand classification models in machine learning, https://docs.microsoft.com/en-us/learn/modules/understand-classification-machine-learning/
- **ML4S ROC**, Measure and optimize model performance with ROC and AUC, https://docs.microsoft.com/en-us/learn/modules/optimize-model-performance-roc-auc/

Supplementary content

- '**XAI**' – learn-tech.io, https://www.learn-tech.io/show-your-workings/
- Excel KNN in a spreadsheet – learn-tech.io
- **DARPA XAI** – https://www.darpa.mil/program/explainable-artificial-intelligence

How can bots alert farmers to weather conditions?

Domain – Data and AI

Big Idea – Learning from Data

Level – 10

SDG – 13. Zero Hunger

Time allocation – 10 hours

Learning Goals

Main

- Evaluate quality, authenticity and accuracy of data and extrapolate from a trend or pattern. Use a range of models and charting methods to analyse, predict and communicate data stories, MCSG
- Clean and prepare textual data for analysis and Machine Learning, MCSG

Supplementary

Turn sources of data into usable text-based services

Concepts

- A bot is a program that runs automatically often on the internet. A bot may 'crawl' for content, engage in conversation ('chatbot') or perform repetitive tasks
- IoT data is data that is harvested from machines, typically through sensors, rather than from human activity
- Useful data is often found in unstructured or semi-structured form such as text and streams of sensor data
- Prescriptive real-time analytics provides up-to-date information along with a recommended course of action

Challenges

Computational Thinking – Extend the Big Question – “How can we connect sensors across schools?” – by writing a program to convert textual weather reports into tabular data useful for machine learning

Data Literacy – Parse IoT data streams and create alerts for anomalous conditions such as extreme winds

Design Thinking – Design a process whereby a farmer could get recommendations on crop management based on parsing of real-time weather reports

Supporting Content

Microsoft content

- **ICJ 2**, Machine Learning, Datasets, Supervised, Unsupervised, Clustering, BOTs, https://imaginecup.microsoft.com/en-us/category/14?skillLevel=0
- **ICJ 3**, Applying AI, Facial recognition and sentiment analysis, NLP, PowerB, https://imaginecup.microsoft.com/en-us/category/14?skillLevel=0
- **ICJ 4**, Deep Learning and Neural Networks, https://imaginecup.microsoft.com/en-us/category/14?skillLevel=0
- **ICJ 5**, AI for Good. AI for Earth, Fishing, Agriculture, Heritage, Humanitarian, https://imaginecup.microsoft.com/en-us/category/14?skillLevel=0
- **ML4S WDA**, Weather data alert, https://docs.microsoft.com/en-gb/learn/modules/send-crop-weather-alerts/

How can ML be used to classify Earth images?

Domain – Data and AI

Big Idea – Learning from Data

Level – 12

SDG – 13. Climate Action

Time allocation – 6:40 hours

Learning Goals

Main

Write a Classical Machine Learning algorithm to classify images, MCSG

Supplementary

- Transform unstructured data into structured data
- Design supervised learning solutions for classification problems

Concepts

- Supervised machine learning uses patterns in data to label things
- Classification problems are those where the label to be learned by the machine is one of a discrete set of classes
- Multinomial logistic regression is a technique used to predict a target variable with more than 2 classes
- A training data set such as ImageNet consists of pre-labelled data used to train a machine learning model
- A confusion matrix is a technique for summarizing the performance of a classification algorithm
- Image classification is the automatic labelling of images by a pre-trained model

Challenges

Computational Thinking – Build a simple image classifier

Data Literacy – Interpret the performance of a machine learning model using a confusion matrix

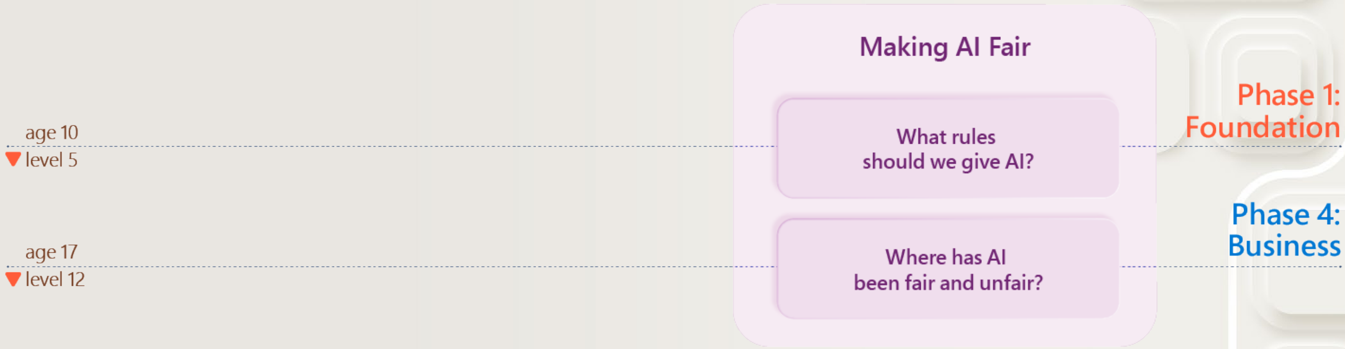
Design Thinking – Design a workflow to train and test an image classification model

Supporting Content

Microsoft content

- **LRN** Class, Create and understand classification models in machine learning, https://docs.microsoft.com/en-us/learn/modules/understand-classification-machine-learning/
- **ML4S CICVS**, Classify images with the Custom Vision service, https://docs.microsoft.com/en-gb/learn/modules/classify-images-custom-vision/
- **ML4S ACDAN**, Analyze climate data with Azure Notebooks, https://docs.microsoft.com/en-gb/learn/modules/analyze-climate-data-with-azure-notebooks/
- **MLCC**: https://aischool.microsoft.com/en-us/machine-learning/learning-paths/ml-crash-course

Making AI Fair



Here the Big Idea focuses on developing the attitudes and mindsets necessary to ensure that the future development of AI is conducted on ethical grounds. Learners will come to understand that AI can be good or bad, but it is controllable. They will appreciate that controlling bias in the data that feeds AI is everyone's responsibility. They will learn to ask probing questions about the predictions that Machine Learning makes and appreciate the importance of XAI in enabling openness and transparency in the production and use of AI, as well as how AI relates to the law.

What rules should we give AI?

Domain – AI

Big Idea – Making AI Fair

Level – 3

Time allocation – 10 hours

Learning Goals

Main

Explain the concepts of ethics, bias and fairness in the context of AI and automation, MCSG

Supplementary

- Develop a baseline understanding of Artificial Intelligence
- Understand that AI can be good or bad, but it is controllable
- Appreciate that controlling bias in data and AI is everyone's responsibility

Concepts

- Leaders and developers have an obligation to make AI ethical
- When you use technology, you are contributing to AI systems
- AI protagonism – control AI or have it done to you

Challenges

Computational Thinking – Write guidance for AI developers to ensure that AI is made ethically

Data Literacy – Explain how the bias of the programmers influence the fairness of the AI rules

Design Thinking – Design a way for people to understand AI fairness rules

Supporting Content

Microsoft content

- **FATE**, Fairness, Accountability, Transparency, and Ethics in AI, https://www.microsoft.com/en-us/research/theme/fate/
- **RAI**, Responsible AI Resources, https://www.microsoft.com/en-us/ai/responsible-ai-resources?activetab=pivot1%3aprimaryr4

Supplementary content

- AI in popular culture
- AI: More Than Human, Barbican, London – https://artsandculture.google.com/project/ai-more-than-human
- Explainable AI 'Show Your Workings': https://vimeo.com/368522294

Where has AI been fair and unfair?

Domain – Data and AI

Big Idea – Making AI Fair

Level – 12

SDG – 16. Peace, Justice and Strong Institutions

Time allocation – 6:40 hours

Learning Goals

Main

Demonstrate a working understanding of XAI and AI Ethics – Design Thinking

Supplementary

- Choose the optimal algorithm to balance performance and transparency for a given problem
- Critically assess the recent history of AI ethics and recommend macro and micro approaches to ensuring AI fairness in the future

Concepts

- Responsible Machine Learning encompasses the following values and principles: understand ML models; protect people and their data; control the end-to-end process
- Explainable AI (XAI) is AI that is designed in such a way that its objective and decision-making processes are transparent and understandable by humans
- Local Interpretable Model Agnostic (LIME) library opens up black box models to examine the driving factors behind each prediction
- Algorithmic bias describes the systemic errors that an algorithm may make which result in unfair outcomes for groups of people, particularly those already marginalized

Challenges

Computational Thinking – Explain the performance and explainability trade-off between different predictive learning algorithms

Data Literacy – Analyse the outputs from an XAI package such as LIME to assess the explainability of a predictive model

Design Thinking – Analyse cases where AI has been clearly unfair, or visibly fair. Design an end-to-end machine learning process that maximizes transparency and ensures fairness

Supporting Content

Microsoft content

- **AI4S**, AI for society, https://www.microsoft.com/en-us/ai/ai-lab-projects#primaryR5
- **GHAI**, Guidelines for Human-AI Interaction, https://www.microsoft.com/en-us/haxtoolkit/ai-guidelines/
- **FL**, Fairlearn, https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/
- **RML**, Responsible Machine Learning, https://docs.microsoft.com/en-us/azure/machine-learning/concept-responsible-ml
- **IncAI**, inclusive AI, https://www.microsoft.com/design/assets/inclusive/InclusiveDesign_InclusiveAI.pdf
- **MIAML**, Model interpretability in Azure Machine Learning, https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability?ocid=AID2463683&wt.mc_id=ai-c9-sejuare

Supplementary content

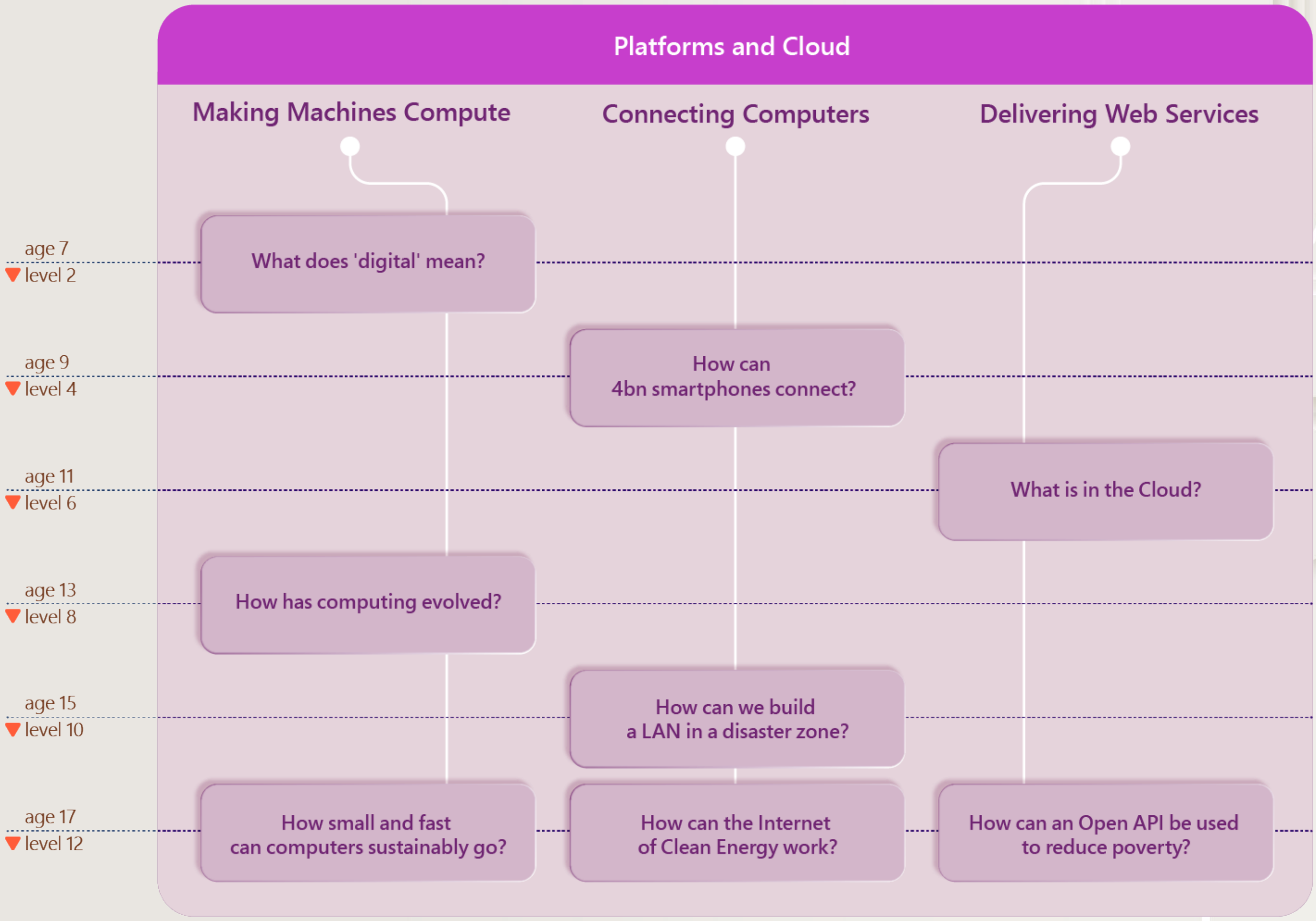
LIME repository: https://github.com/marcotcr/lime

Platforms and Cloud

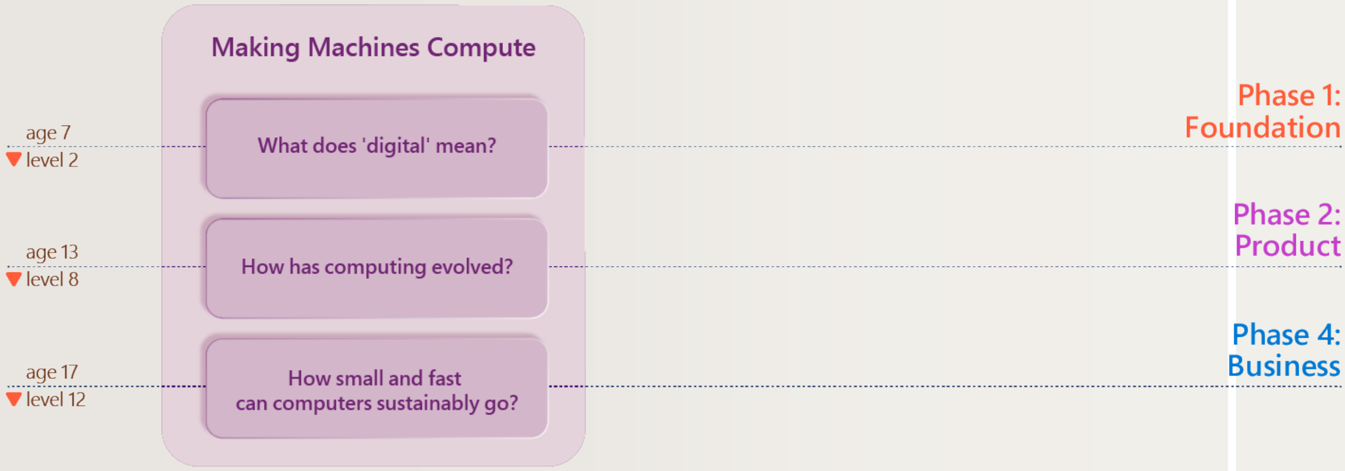
A platform is the environment in which a piece of software is executed, and a group of technologies that are used as a base upon which other applications, processes or technologies are developed. Anyone wishing to write software for broad use needs to understand how to work with platforms.

The goal of this Domain is to develop skills in working with platforms by using the following Big Ideas:

- Making Machines Compute
- Connecting Computers
- Delivering Web Services


Platforms and Cloud
Making Machines Compute
Connecting Computers
Delivering Web Services
age 7
level 2
What does 'digital' mean?
age 9
level 4
How can 4bn smartphones connect?
age 11
level 6
What is in the Cloud?
age 13
level 8
How has computing evolved?
age 15
level 10
How can we build a LAN in a disaster zone?
age 17
level 12
How small and fast can computers sustainably go?
How can the Internet of Clean Energy work?
How can an Open API be used to reduce poverty?

Making Machines Compute



This Big Idea explores how hardware works, the components that different platforms are based on and how computing scales are measured. Here students explore the interface between electronics, physics, and computing to understand how code gets executed.

Students will learn that computing exists on a spectrum, at one end of which are embedded devices (SOCs), and at the opposite end are supercomputers used for running massive mathematical models. Content will cover areas such as the units used to measure computing scales, and the consumption and conservation of energy in delivering different types of computing services.

What does 'digital' mean?

Domain – Platforms

Big Idea – Making Machines Compute

Level – 2

Time allocation – 10 hours

Learning Goals

Main

- Use appropriate terminology in identifying and describing the function of common physical components of computing systems, CSTA 1A-CS-02
- Know that digital computers use binary to represent all data, how binary can represent numbers and images, computers transfer data in binary, relationship between binary and file size, CAS D&DR 5,

Supplementary

Develop basic block-based coding skills

Concepts

- The term 'digital' means data expressed as series of the digits 0 and 1, and refers to elements such as hardware, software and networks
- Digital hardware and software components can be used to transform data into digital solutions
- Programs are written in high level (close to English) source code, which is then compiled into an 'executable' format in binary
- Executable code is activated in an operating system (e.g. Word executable code is activated in Windows)

Challenges

Computational Thinking

- Use a computer to search for information.
- Describe the different parts and types of the computer and their functions, and the difference between operating systems and applications and their functions

Data Literacy – Program a micro:bit watch to count and display the number of movements you make

Design Thinking – Make a movement counter 'watch' with a micro:bit

Supporting Content

Microsoft content

- **MC CSP AP 1**, Computer Science Principles course, Unit 1 – Introduction to AP Computer Science Principles, https://microsoft.github.io/makecode-csp/unit-1/introduction
- **MC CC**, Coding Cards, https://makecode.microbit.org Coding Cards
- **MC HW**, Hardware, https://makecode.microbit.org – Behind the MakeCode Hardware
- **MC Tut**, Tutorials, https://makecode.microbit.org Tutorials for Micro:Bit V2
- **MC CSP AP 3&5**, Computer Science Principles course, Units 3 & 5 Algorithms and Programming Part 1 & 2, https://microsoft.github.io/makecode-csp/unit-3/algorithms-programming & https://microsoft.github.io/makecode-csp/unit-5/algorithms-programming
- **MC** Watch, MakeCode micro:bit Watch project, https://makecode.microbit.org/projects/watch

Supplementary content

- Pearson, Digital Technology Demystified – https://qualifications.pearson.com/en/about-us/qualification-brands/btec-international/learning-hub/artificial-intelligence/digital-technology-de-mystified.html

How has computing evolved?

Domain – Platforms

Big Idea – Making Machines Compute

Level – 8

Time allocation – 10 hours

Learning Goals

Main

- Describe how internal and external parts of computing devices function to form a system, CSTA 1B-CS-01
- Understand the main functions of the operating system, CAS H&P 4

Supplementary

- Understand how a chip receives readable code to process inputs into outputs
- Understand the fundamental components of a computer
- Understand that while the form of computing changes, the essential aspects persist

Concepts

- The powerful computer then and now (memory/storage/processor/input/output)
- On-board computers controlling the space rocket
- Computer calculations for trajectory (Mission Control)

Challenges

Computational Thinking

- Gain a working understanding of the Raspberry Pi
- Explain the similarities and differences between the NASA 1969 computer and modern computers
- Compare the features of a Raspberry Pi to the computers used by NASA in 1969

Data Literacy – Explain how data inputs are stored and accessed in memory

Design Thinking

- Design a program that set a rocket's course for the moon
- Design a science experiment that uses the GPIO functions in a Raspberry Pi

Supporting Content

Microsoft content

- RPi, Setting up a Raspberry Pi, https://docs.microsoft.com/en-us/windows/iot-core/tutorials/rpi

Supplementary content

- Intel, "The Journey Inside", Introduction to Computers (Intel 1-6) – https://www.intel.com/content/www/us/en/education/k12/the-journey-inside/explore-the-curriculum/intro-to-computers.html

- https://www.realclearscience.com/articles/2019/07/02/your_mobile_phone_vs_apollo_11s_guidance_computer_111026.html
- https://en.wikipedia.org/wiki/Apollo_Guidance_Computer
- https://www.youtube.com/watch?v=g0PIXvjTasI
- https://spaceplace.nasa.gov/dsn-antennas/en/
- https://www.pbs.org/video/how-computers-calculate-the-alu-crash-course-computer-sci-sm5zov/
- https://www.learneroo.com/modules/12/nodes/81
- https://spaceplace.nasa.gov/build-a-spacecraft/en/d

How small and fast can computers sustainably go?

Domain – Platforms & Cloud

Big Idea – Making Machines Compute

Level – 12

SDG – 13. Climate Action

Time allocation – 6:40 hours

Learning Goals

Main

- Demonstrate a working understanding of the criticality of factoring energy use into a solution, MCSG
- Demonstrate a basic understanding of quantum computing?

Supplementary

- Approach sustainable computing problems from several angles including platform choice, computer design and software design
- Write a simple program for a quantum computer

Concepts

- Moore's Law is the postulation that silicon chip-based computers will double their power density every year.
- An integrated circuit (IC) is a small silicon chip that can function as an amplifier, oscillator, timer, microprocessor, or computer memory
- Cloud computing may save energy through resource virtualization and higher utilization although increased demand may cause a net increase in energy usage
- A quantum computer is one that leverages the unique behaviours of subatomic particles (such as entanglement) to process information in terms of probabilities rather than binary states.
- Energy proportionality is a measure of the relationship between power consumed in a computer system, and the rate at which useful work is done
- We use Big O notation to describe the time complexity of an algorithm from which we can infer its energy requirements.

Challenges

Computational Thinking – Code a quantum circuit using a quantum development kit

Data Literacy – Work out how to minimize the energy costs of a program across multiple platform, computer and software implementations

Design Thinking – Design a new low-powered, wearable computing device

Supporting Content

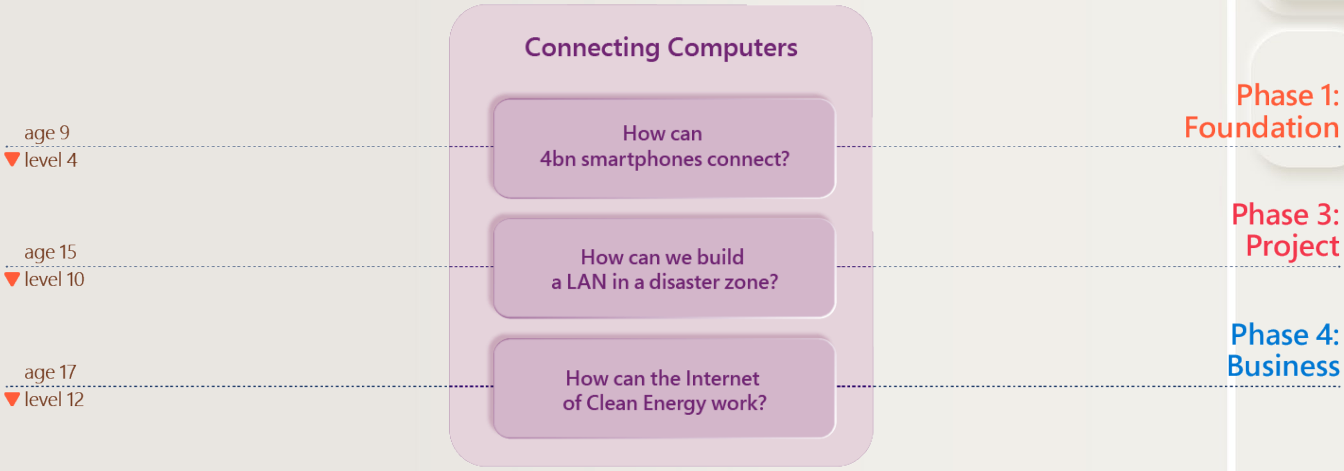
Microsoft content

- IQ, Introduction to Quantum Computing, https://docs.microsoft.com/en-us/quantum/overview/overview

Supplementary content

- Introduction to Quantum Computing: https://www.ibm.com/quantum-computing/

Connecting Computers


Connecting Computers
age 9
level 4
How can 4bn smartphones connect?
Phase 1: Foundation
age 15
level 10
How can we build a LAN in a disaster zone?
Phase 3: Project
age 17
level 12
How can the Internet of Clean Energy work?
Phase 4: Business

The Internet is something we tend to take for granted, but an understanding of its mechanics is essential to Computer Science. In this Big Idea we explore how the hardware, software, protocols combine to make networks and mobile platforms work. Another key component of this Big Idea is the API and how it can be used in software development.

How can 4bn smartphones connect?

Domain – Platforms

Idea – Platforms and Cloud

Level – 2

Time allocation – 10 hours

Learning Goals

Main

- Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination, Design Thinking, CSTA 1B-NI-04
- Know that digital computers use binary to represent all data, how binary can represent numbers and images, computers transfer data in binary, relationship between binary and file size, CAS D&DR 5

Supplementary

- Gain an appreciation of the rich functionality of the Internet from entertainment to businesses to education
- Understand the Internet as a large collection of connected digital devices and information

Concepts

- When digital systems are connected, they form a network
- The Internet is a physical entity made of computers, connections and protocols
- Apps on your phone exchange data over the internet
- The Internet is a platform for providing social value e.g. helping people find each other after an earthquake, or for doctor to perform a surgery remotely

Challenges

Computational Thinking – Identify connected devices. Connect micro:bits with wires and with Bluetooth, and transfer data between them

Data Literacy – Binary numbers, image representation

Design Thinking – Model a packet's journey. Draw a map of the behind-the-scenes connections when you use WhatsApp, showing components and scales

Supporting Content

Microsoft content

- **H-STEM HEC**, Harnessing Electricity to Communicate, https://www.microsoft.com/en-us/education/education-workshop/telegraph.aspx
- **MC CSP AP 2**, Computer Science Principles course, Unit 2 -Computing Systems and Networks, https://microsoft.github.io/makecode-csp/unit-2/computing-systems-networks
- **MC Com1**, Serial data, https://makecode.microbit.org/device/serial
- **MC Com2**, Bluetooth pairing, https://makecode.microbit.org/reference/bluetooth/bluetooth-pairing
- **MDL WWC 5**, Working With Computers, Getting Online, https://www.linkedin.com/learning/working-with-computers-and-devices/working-with-computers-to-create-opportunity?autoAdvance=true&autoSkip=false&autoplay=true&resume=true
- **MDL WCOL**, Working and Collaborating Online – WiFi, safety and responsible behaviour, video calls, file sharing, https://www.linkedin.com/learning/working-and-collaborating-online/computing-with-confidence-getting-job-skills-and-staying-safe-online?autoAdvance=true&autoSkip=false&autoplay=true&resume=true
- **MDL LC2**, English Localised Course 2 – Access information online, videos, connect, browse, evaluate, https://www.microsoft.com/en-us/digital-literacy
- **MDL LC3**, English Localised Course 3 – Communicate online, videos, email, chat and video calling, https://www.microsoft.com/en-us/digital-literacy
- **MDL LC5**, English Localised Course 5 – Create digital content, files, text, pictures, tables, printing, https://www.microsoft.com/en-us/digital-literacy
- **MDL LC6**, English h Localised Course 6 – Collaborate and manage content digitally, https://www.microsoft.com/en-us/digital-literacy

Supplementary content

- Connecting micro:bits with wires – https://microbit.nominetresearch.uk/networking-book-online/wiredcommunication/wiredcommunication/
- Computer Science Unplugged (CSU 1) Binary Numbers https://csunplugged.org/en/topics/binary-numbers/
- CSU 3 Image Representation https://csunplugged.org/en/topics/image-representation/
- Binary https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/using-binary-to-create-on-off-pictures
- There and Back Again: A Packet's Tale, https://www.youtube.com/watch?v=ewrBalT_eBM

- https://edu.gcfglobal.org/en/internetbasics/what-is-the-internet/1/
- https://www.monitis.com/blog/what-will-the-future-internet-look-like/

How can we build a LAN in a disaster zone?

Domain – Platforms

Idea – Connecting Computers

Level – 10

SDG – 3. Good Health and Well-Being

Time allocation – 10 hours

Learning Goals

- Demonstrate an understanding of the relative merits of different network methods, the role of Internet protocols and how packets, IP addresses, and memory works, MCSG
- Understand data transmission between digital computers over networks, including the internet i.e. IP addresses and packet switching, CAS C&N 5
- Explain the names of hardware e.g. hubs, routers, switches, and the names of protocols e.g. SMTP, iMAP, POP, FTP, TCP/IP, associated with networking computer systems, CAS C&N 5

Concepts

- Networks are architected in layers from device to Local Area Networks to the Internet and back to devices
- Peer-to-Peer Networks share work and privileges across all nodes (computers) in the network
- A Client/Server Networks is one where a server hosts resources accessed by clients (service requestors) and, in the case of local networks, may control the clients
- A low-power wide area network for low-powered edge devices (e.g. LORA) can provide rescue teams communication even in the absence of a cellular network

Challenges

Computational Thinking

- Set up a physical or virtual network to manage and utilize limited resources
- Set up policies that manage people and resources
- Model ways to bring hardware such as sensors and cameras into the network

Data Literacy – Manage data access across a network

Design Thinking – Draw a diagram for a low energy network of devices that could be used to robustly transmit data to and from a disaster zone

Supporting Content

Microsoft content

- **FCN**, Fundamentals of Computer Networking, https://docs.microsoft.com/en-us/learn/modules/network-fundamentals/
- **ML4S PCC**, Principles of cloud computing, https://docs.microsoft.com/en-gb/learn/modules/principles-cloud-computing
- **ML4S CAA**, Create an Azure account, https://docs.microsoft.com/en-gb/learn/modules/create-an-azure-account/

- **QS-AVN**, Quickstart: Create a virtual network using the Azure portal, https://docs.microsoft.com/en-us/azure/virtual-network/quick-create-portal

Supplementary content

- https://www.mdpi.com/2227-7080/5/4/65/pdf
- https://ieeexplore.ieee.org/document/8647177

How can the Internet of Clean Energy work?

Domain – Platforms & Cloud

Idea – Connecting Computers

Level – 12

SDG – 7. Affordable Clean Energy

Time allocation – 6:40 hours

Learning Goals

Main

Parameterise a Neural Network to solve a predictive modelling problem, MCSG

Supplementary

- Demonstrate understanding of the key concepts of neural network architectures
- Map appropriate software and hardware to complex problems

Concepts

- Intermittent production from renewable energy sources creates a challenge to balance energy supply and demand
- Root Mean Squared Error (RMSE) is a key measure of model performance for a prediction problem
- A neural network is a machine learning architecture inspired by biological neurons that can be used for predictions.
- Deep neural networks (Deep Learning) use many layers of neurons to enhance model performance
- Tensorflow is a powerful deep learning library available in Python, C and other programming languages

Keras is a Python API that provides an intuitive interface to powerful Deep Learning solvers

Challenges

Computational Thinking – Train a deep neural network to predict energy consumption

Data Literacy – Calculate and interpret performance metrics for regression outputs

Design Thinking – Design a solution to help manage an energy grid containing clean power generation

Supporting Content

Microsoft content

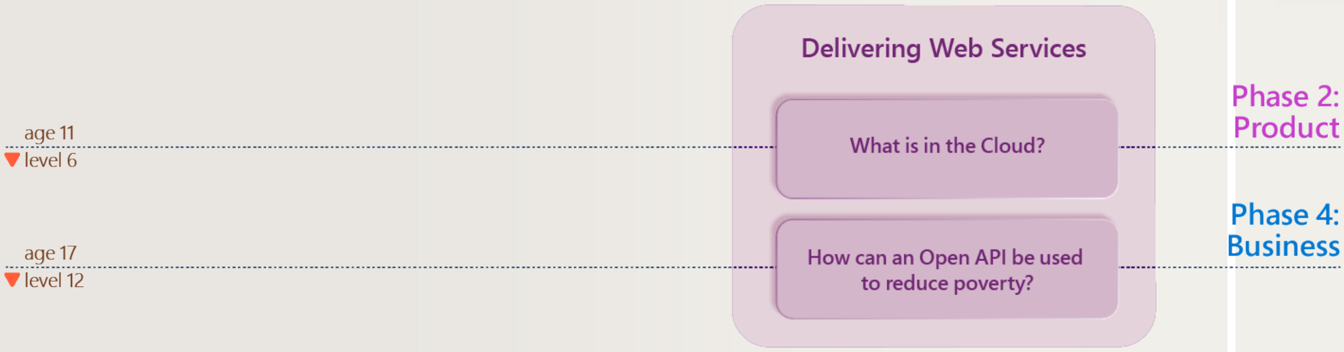
- **ML4S CO**, Compute options, https://docs.microsoft.com/en-gb/learn/modules/intro-to-azure-compute/
- **ML4S SO**, Storage options, https://docs.microsoft.com/en-gb/learn/modules/intro-to-data-in-azure/
- **ML4S ASG**, Architecture and service guarantees, https://docs.microsoft.com/en-gb/learn/modules/explore-azure-infrastructure/

- **ML4S I2AIoT**, Introduction To Azure IoT, https://docs.microsoft.com/en-us/learn/paths/introduction-to-azure-iot/
- **IDL**, Intro to Deep Learning, https://aischool.microsoft.com/en-us/machine-learning/learning-paths/build-a-predictive-maintenance-solution-using-deep-learning
- **ML4S I2NN**, Introduction to Neural Networks, https://docs.microsoft.com/en-gb/learn/modules/introduction-to-neural-networks/
- **NNR**, Neural Network Regression, https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/neural-network-regression

Supplementary content

- Internet of Clean Energy – https://www.learn-tech.io/internet-of-clean-energy/

Delivering Web Services



Here learners are introduced to the concept of the Cloud and given an understanding of how it can be used to deliver services. This Big Idea allows learners to appreciate the links between traditional CS concepts, such a data storage, with modern concepts such as Big Data and virtualization.

Equipped with an understanding of technology "stacks", students can then turn their attention to authentic problem-based learning projects based on SDG goals, exploring how an Open API can be used to reduce poverty, for example.

What is in the Cloud?

Domain – Platforms and Cloud

Big Idea – Delivering Web Services

Level – 6

Time allocation – 10 hours

Learning Goals

Main

- Understand the concept of 'stack'
- Arrange elements of a Cloud-based stack, including a database, in a diagram, MCSG
- Recognise and understands the function of the main internal parts of basic computer architecture, CAS H&P 5
- Use units and scales of computing, MCSG

Supplementary

- Understand the outsourcing of work from local computers to the Cloud
- Understand the services available on the cloud (SAAS – Software as a Service)
- Learn how citizens can manage their data and usage of the Cloud

Concepts

- Cloud storage
- Cloud computing
- Software as a Service (SAAS)

Challenges

Computational Thinking – Use and explain the Cloud

Data Literacy – Explain the physical data storage that sits behind commonly used Cloud services

Design Thinking – Draw what happens in the Cloud when you use a service such as Facebook

Supporting Content

Microsoft content

- **ML4S NO**, Networking options, https://docs.microsoft.com/en-gb/learn/modules/intro-to-azure-networking/
- **ML4S MSAP**, Manage services with the Azure portal, https://docs.microsoft.com/en-gb/learn/modules/tour-azure-portal/
- **ACWA**, Azure Create a Web App, https://azure.microsoft.com/en-us/get-started/web-app/
- **ML4S WPF**, Build an AI web app by using Python and Flask, https://docs.microsoft.com/en-us/learn/modules/python-flask-build-ai-web-app/
- **ML4S AAIAP**, Apply and monitor infrastructure standards with Azure Policy, https://docs.microsoft.com/en-gb/learn/modules/intro-to-governance/

Supplementary content

- https://kids.kiddle.co/Cloud_computing
- https://www.youtube.com/watch?v=TTNgV0O_oTg
- https://www.youtube.com/watch?v=ZnGzxV_dQeg

How can an Open API be used to reduce poverty?

Domain – Platforms and Cloud

Idea – Delivering Web Services

Level – 12

SDG – 1. No Poverty

Time allocation – 6:40 hours

Learning Goals

Main

Write code to ingest data from a public API, MCSG

Supplementary

- Use functionality provided through a third-party API in a program
- Create and deploy a simple Web API

Concepts

- A web service is a software application (resource) that is made available on the internet
- A Web API (Application Programming Interface) is a set of standards and protocols that allow a programmer to interact with 3rd party internet resources such as data or functionality
- An open API is a Web API that is made available to the public
- APIs typically exchange data as JSON (JavaScript Object Notation) files

Challenges

Computational Thinking – Write a program to ingest, process and store data from various publicly available APIs

Data Literacy – Manipulate incoming data and process it into useful summary tables in the database

Design Thinking – Using publicly available APIs, design a banking application for people in the developing world who currently don't have a bank account

Supporting Content

Microsoft content

- **ML4S CMAX**, Create Mobile App with Xmarin, https://docs.microsoft.com/en-gb/learn/modules/create-a-mobile-app-with-xamarin-forms/

Human-Computer Interaction

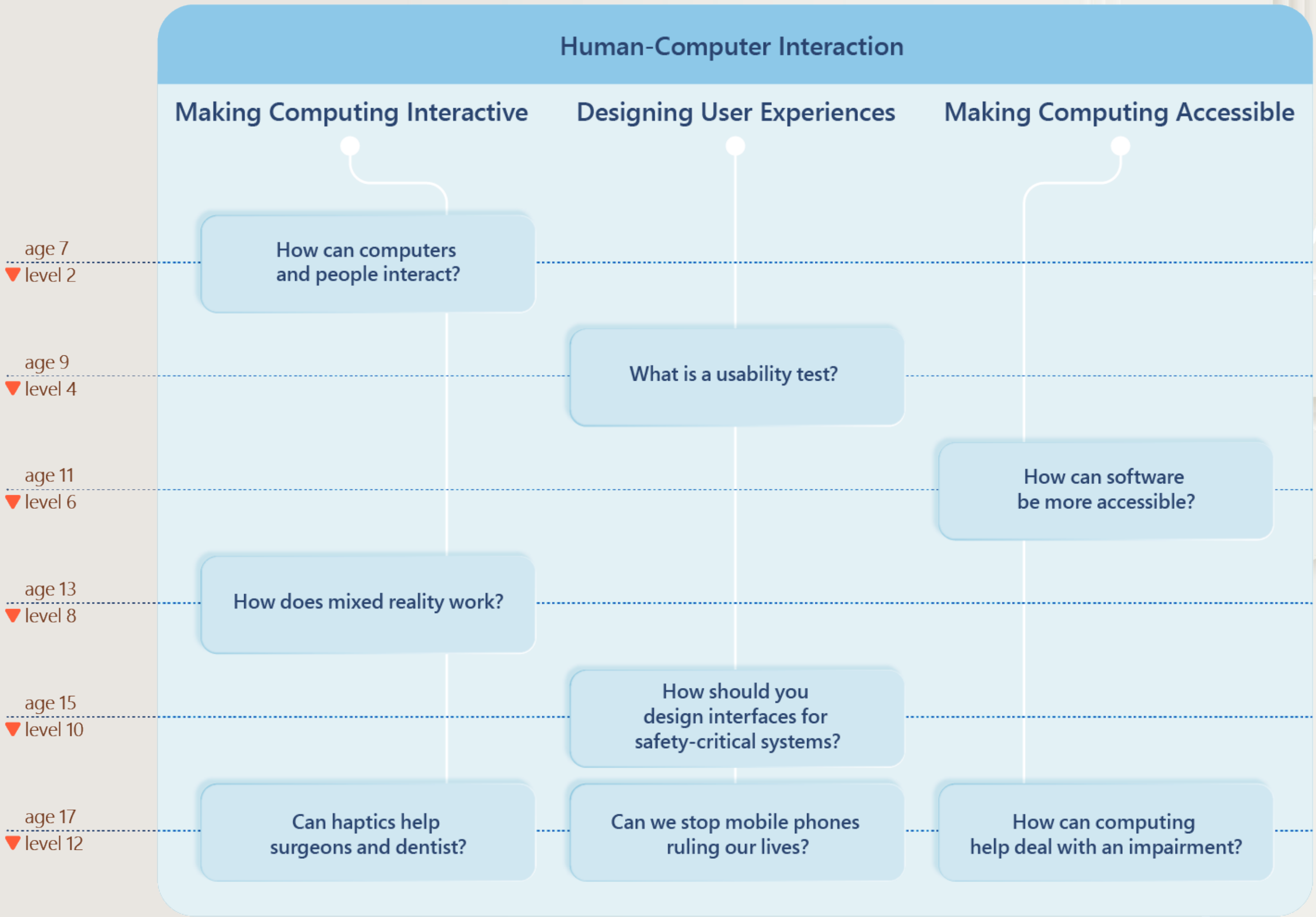
The ways in which people can interact with computing has expanded rapidly in recent times and this trend is set to continue. In the Knowledge Era human computer interaction (HCI) was almost exclusively by screen, keyboard and mouse. Now, in the Artificial Intelligence Era, we have broadening ways to interact with computing, for example voice, gesture, haptics, mixed reality (MR) and sentiment.

As computing broadens and proliferates, a key factor that determines the success of a computing solution is how well it is designed in terms of the user experience (UX). As interaction with computing becomes increasingly interwoven with everyday life, it is also essential that developers understand how to make their technology accessible to all.

An understanding of how to design HCI is therefore an essential asset for anyone designing and building computing solutions.

The goal of this Domain is to develop HCI design skills by using the following Big Ideas:

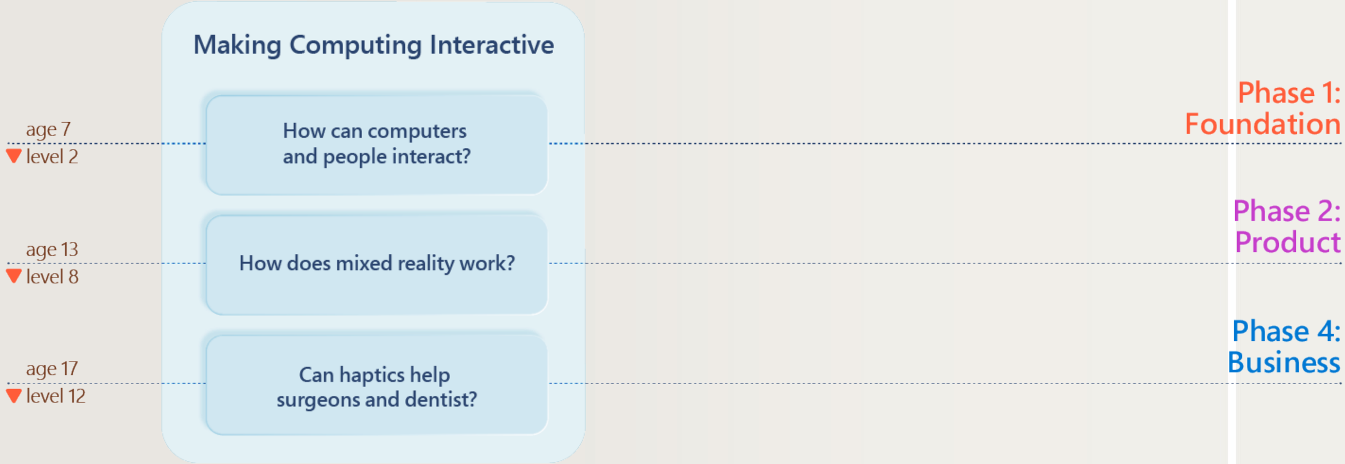
- Making Computing Interactive
- Designing User Experiences
- Making Computing Accessible



Human-Computer Interaction

	Making Computing Interactive	Designing User Experiences	Making Computing Accessible
age 7 ▼ level 2	How can computers and people interact?		
age 9 ▼ level 4		What is a usability test?	
age 11 ▼ level 6			How can software be more accessible?
age 13 ▼ level 8	How does mixed reality work?		
age 15 ▼ level 10		How should you design interfaces for safety-critical systems?	
age 17 ▼ level 12	Can haptics help surgeons and dentist?	Can we stop mobile phones ruling our lives?	How can computing help deal with an impairment?

Making Computing Interactive



	Making Computing Interactive	
age 7 ▼ level 2	How can computers and people interact?	Phase 1: Foundation
age 13 ▼ level 8	How does mixed reality work?	Phase 2: Product
age 17 ▼ level 12	Can haptics help surgeons and dentist?	Phase 4: Business

The key goal of this Big Idea is for students to understand how key HCI technologies work from a scientific point of view. For example, how exactly does finger swipe on a screen trigger an action, or how exactly can voice control a computing service? Knowing the capabilities and limitations of HCIs such as haptics or tracking is an important capability for aspiring computer scientists to have.

How can computers and people interact?

Domain – Human-Computer Interaction

Big Idea – Making Computing Interactive

Level – 2

Time allocation – 10 hours

Learning Goals

Main

- Explain the relative merits of different types of interface, MCSG
- Seek diverse perspectives for the purpose of improving computational artefacts, CSTA 1B-IC-20
- Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use – Design Thinking, CSTA 1A-CS-01
- Develop programs with sequences and simple loops, to express ideas or address a problem – Computational Thinking, CSTA 1A-AP-10

Supplementary

- Have an understanding of the word "interface"
- Create digital content, interacting with text, pictures, lists and other types of objects
- Describe what coding is
- Know simple programming commands
- Create interactive stories and games using graphical programming blocks

Concepts

- People use computing devices to perform a variety of tasks and should select the appropriate app/program to complete these.
- A narrative can be weaved into a program with tools such as speech bubbles, sound and movement commands
- We can set our own goals and develop individualised interactions with computing applications
- Open learning environments such as Minecraft give us the freedom to experiment, encouraging creative self-expression and problem solving

Challenges

Computational Thinking – Write programs with sequences and loops

Data Literacy – Collect data through surveys and interpret results using a program

Design Thinking **–** Propose ways in which software user interfaces could be improved

Supporting Content

Microsoft content

- **MDL WWC 2**, Working With Computers, Working with Destkop OS, https://www.linkedin.com/learning/working-with-computers-and-devices/working-with-computers-to-create-opportunity?autoAdvance=true&autoSkip=false&autoplay=true&resume=true

- **MDL WWC 3**, Working With Computers, Working with Applications, https://www.linkedin.com/learning/working-with-computers-and-devices/working-with-computers-to-create-opportunity?autoAdvance=true&autoSkip=false&autoplay=true&resume=true
- **MEE CWM 1**, Computing WIth Minecraft, The Agency, Code an Agent to move in different directions, https://education.minecraft.net/en-us/lessons/computing-the-agency

Supplementary content

- Code.org, (CSD Unit 6), Fitting coding to screens – https://curriculum.code.org/csd-1718/unit6/2/
- CSG, HCI – https://csfieldguide.org.nz/en/chapters/human-computer-interaction/

How does Mixed Reality (MR) work?

Domain – Human-Computer Interaction

Big Idea – Making Computing Interactive

Level – 8

Time allocation – 10 hours

Learning Goals

Main

Use reverse engineering to understand a solution, MCSG

Supplementary

- Understand how virtual and augmented reality (XR) work
- Learn how to work with mixed reality systems

Concepts

- However impressive a piece of technology, it can be explained by engineering and technical principles
- 6DoF (six degrees of freedom), pitch, yaw and roll
- The need for low latency for XR to be effective
- Understand what a hologram is

Challenges

Computational Thinking

- Take apart a VR headset and analyse the components
- Explain how the headset recreates a real world-type environment
- Build an e-fashion item that displays head tracking data
- Explain how eye tracking works

Data Literacy – Explain how objects can be defined in three-dimensional space

Design Thinking

- Produce content for a mixed reality environment
- Research and explain use cases where MR is an appropriate part of a solution

Supporting Content

Microsoft content

- **ML4S I2MR**, Introduction to mixed reality, https://docs.microsoft.com/en-us/learn/modules/intro-to-mixed-reality/
- **ML4S Holo, HoloLens 2** fundamentals, https://docs.microsoft.com/en-us/learn/paths/beginner-hololens-2-tutorials/
- **ML4S BCMR**, Prepare to bring the basketball court into a Mixed Reality space, https://docs.microsoft.com/en-us/learn/modules/mixed-reality-for-sports-fans/

Can haptics help surgeons and dentist?

Domain – Human Computer Interaction

Idea – Making Computing Interactive

Level – 12

SDG – 3. Good Health and Well-Being

Time allocation – 6:40 hours

Learning Goals

Main

Have a working understanding of haptics, tracking and other advanced HCI technology, MCSG

Supplementary

- Conceptualize and design multimodal interactive systems
- Write a simple program for non-traditional interactivity with the user

Concepts

- Haptics refers to the use of technology that simulates the senses of touch and motion, especially to simulate a physical sensation from a remote connection
- Telesurgery utilizes wireless networking and robotic technology to connect surgeons and patients who are distantly located from one another
- Multimodal interactive systems allow the user to interact with a computer through multiple modes such as touch, gestures, speech or keyboard input
- 5G (fifth generation) mobile networks have increased speed and coverage and reduced latency which allows real-time control of IoT devices even in hazardous situations
- Material Design is a visual language that synthesizes the classic principles of good design with the innovation of technology and science

Challenges

Computational Thinking – Write a program designed to use haptic inputs and outputs

Data Literacy – Analyse the information transmission requirements for remote surgery in terms of bandwidth and latency

Design Thinking – Design the user experience (UX) for a surgeon who will undertake remote surgery

Supporting Content

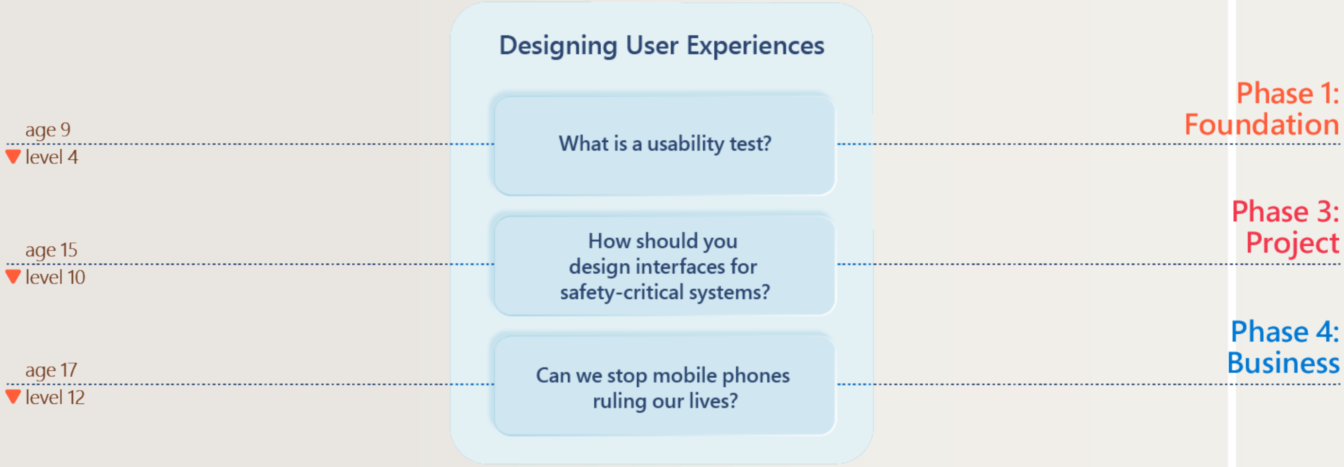
Microsoft content

- **LRN IoT Sol**, Examine the architecture of an IoT solution, https://docs.microsoft.com/en-us/learn/modules/introduction-iot-solution-architecture/
- **LRN IoT Cent**, Intro to IoT Central, https://docs.microsoft.com/en-us/learn/modules/examine-azure-iot-central-application-architecture/
- **GSMR**, Get started with Mixed Reality, https://docs.microsoft.com/en-us/windows/mixed-reality/

Supplementary content

- Medium (MTRS), Ericsson and King's College London demonstrate 5G tactile robotic surgery – https://www.ericsson.com/en/press-releases/2016/6/ericsson-and-kings-college-london-demonstrate-5g-tactile-robotic-surgery
- Material Design for Haptics: https://material.io/design/platform-guidance/android-haptics.html
- https://www.computer.org/csdl/magazine/co/2012/04/mco2012040021/13rRUxBJhq2
- https://medium.muz.li/haptic-ux-the-design-guide-for-building-touch-experiences-84639aa4a1b8?gi=36abc49f83b

Designing User Experiences


Designing User Experiences

What is a usability test?

How should you design interfaces for safety-critical systems?

Can we stop mobile phones ruling our lives?

Ultimately there is little point in learning CS unless people are going to use your products. Therefore, this Big Idea is about developing skills in usability testing, user-centric design, and interface design.

What is a usability test?

Domain – Human – Computer Interaction

Big Idea – Designing User Experiences

Level – 4

Time allocation – 10 hours

Learning Goals

Main

- Think of ways to improve the accessibility and usability of computing, MCSG

- Evaluate the trustworthiness of digital content and consider the usability of visual design features when designing and creating digital artefacts for a known audience**,** CAS IT 6
- Recommend improvements to the design of computing devices, based on analysis of how users interact with the devices, CSTA – 2-CS-01

Supplementary

- Be aware of the need to consider different perspectives, ability levels, points of view, and disabilities in the development of inclusive technology
- Recognise that collaboration and sharing of ideas allow the benefit of diverse perspective

Concepts

- Human-computer interaction (HCI) is about trying to make programs useful, usable, and accessible to humans
- Evaluating interface is best done by getting feedback from lots of potential users
- Usability heuristics in user interface design
- User Experience (UX) is about interactions with the device or application, and the aim is to make a product intuitive

Challenges

Computational Thinking – Design and explain an app that uses UX design principles

Data Literacy – Use focus groups or surveys and the Statistical Investigation Process to evaluate usability

Design Thinking – Identify users and considerations that may be significant, evaluate interfaces using think aloud protocols and the cognitive walkthrough

Supporting Content

Microsoft content

- **ML4S ISTC**, Introduction to software testing concepts, https://docs.microsoft.com/en-us/learn/modules/visual-studio-test-concepts/
- **MC GJIAB** 2021, Globald Game Jam, Arcade 2021, game design, Tilemaps, Sprites, HUD (Head-up Display), Status Bar, Art and Design, Animations, Effects, https://arcade.makecode.com/gamejam/global-2021
- **MC GJIAB**, Game Jam in a Box, multiple Game Jams, competition for participants to make and showcase games, Game design, Tilemaps, Sprites, HUD (Head-up Display), Status Bar, Art and Design, Animations, Effects, https://arcade.makecode.com/gamejam/lessons/box

Supplementary content

- Code.org (CSD Unit 4) – The Design Process, User-Centred Design, UX, prototyping, testing – https://studio.code.org/s/csd4-2019
- Code.org CSD Unit 6 – Physical Computing. Fitting coding to screens and using virtual sensor data https://studio.code.org/s/csd6-2019
- CS Field Guide (NZ), (CSFG 1), Human-Computer Interaction https://csfieldguide.org.nz/en/chapters/human-computer-interaction/
- Jakob Nielsen, Usability Heuristics https://www.nngroup.com/articles/ten-usability-heuristics/
- Statistical Investigation Process https://new.censusatschool.org.nz/resource/statistical-investigation/

How should you design interfaces for safety-critical systems?

Domain – Human-Computer Interaction

Big Idea – Designing User Experiences

Level – 11

SDG – 3. Good Health and Well-Being

Time allocation – 10 hours

Learning Goals

Main

- Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices, CSTA 2-CS-01
- Apply principles of UX design, MCSG
- Understand the importance of fail-safe and zero errors in safety-critical systems, MCSG

Concepts

- Safety-critical systems are those systems whose failure could result in loss of life, significant property damage, or damage to the environment.
- Aircraft, cars, weapons systems, medical devices, and nuclear power plants are the traditional examples of safety-critical software systems.
- Software developed for these types of use cases need to be certified as safety-critical
- Developers need to write software and design interfaces to take account of safety
- Safety-critical software development is a methodical, process-driven field of software development

Challenges

Computational Thinking

- Produce a set of rules for testing and certifying safety-critical software
- Build a prototype interface for a safety-critical system

Data Literacy – Use statistical methods to assess the effectiveness of a safety-critical interface design

Design Thinking – Design interfaces for a safety-critical system

Supporting Content

Microsoft content

- **ML4S ADTU**, Set up a mixed reality project for Azure Digital Twins in Unity, https://docs.microsoft.com/en-us/learn/modules/set-up-mixed-reality-azure-digital-twins-unity/
- **ML4S MRADTU**, Build mixed reality digital twins with Azure Digital Twins and Unity, https://docs.microsoft.com/en-us/learn/paths/build-mixed-reality-azure-digital-twins-unity/
- **ML4S ISTC**, Introduction to software testing concepts, https://docs.microsoft.com/en-us/learn/modules/visual-studio-test-concepts/
- **UNTY**, Unity Tutorials, https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/tutorials
- **ML4S 3dU**, Build a 3D Scene for mixed reality in Unity, https://docs.microsoft.com/en-us/learn/modules/build-3d-scene-mixed-reality-unity/
- **ML4S 3dAfU**, Configure 3D assets for mixed reality in Unity, https://docs.microsoft.com/en-us/learn/modules/configure-3d-assets-mixed-reality-unity/

Supplementary content

- NASA Software Safety Guidebook – https://standards.nasa.gov/standard/nasa/nasa-gb-871913
- Safety Critical Software – https://smallbusinessprogramming.com/safety-critical-software-15-things-every-developer-should-know/

Can we stop mobile phones ruling our lives?

Domain – Human Computer Interaction

Idea – Designing User Experiences

Level – 12

SDG – 9. Industry, Innovation and Infrastructure

Time allocation – 6:40 hours

Learning Goals

Main

Use user-centric design methods, *Design Thinking, MCSG*

Supplementary

- Design an application that prioritizes human and societal needs
- Write a simple mobile application

Concepts

- Time Well Spent is a human-centric measure of application usage
- Platforms such as Xmarin for iOS and Android facilitate development and deployment of mobile apps
- Open source design frameworks such as Fluent Design System facilitate design of mobile apps

Challenges

Computational Thinking – Develop and deploy a simple mobile application that 'nudges' mobile phone behaviour towards less but better-quality use

Data Literacy – Understand the dataflows in and out of your app

Design Thinking – Design the app so that it will foster mental wellness

Supporting Content

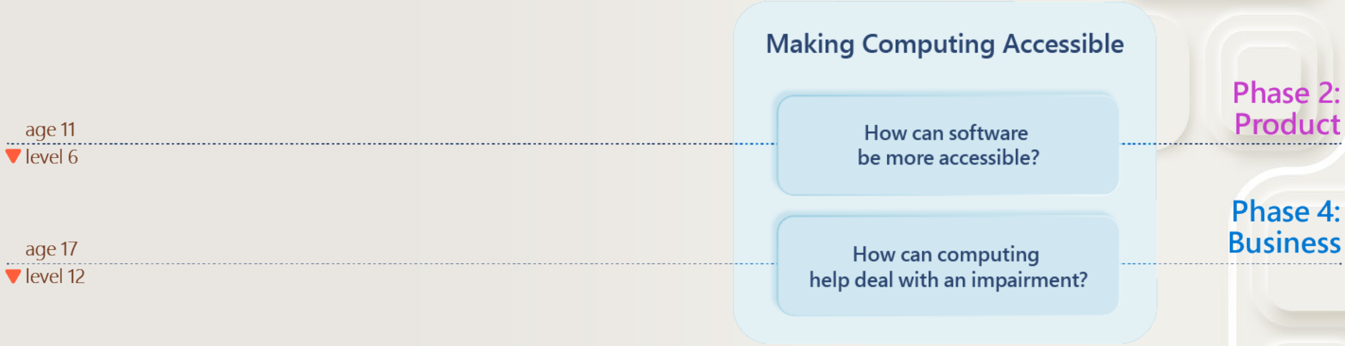
Microsoft content

- **LRN** Xamarin, Create a mobile app with Xamarin.Forms, https://docs.microsoft.com/en-us/learn/modules/create-a-mobile-app-with-xamarin-forms/
- **MSD**, Microsoft Design – https://www.microsoft.com/design
- **FL**, Fluent, Microsoft Fluent Design System, https://www.microsoft.com/design/fluent/

Supplementary content

- https://learnappmaking.com/how-to-make-an-app/
- https://buildfire.com/how-to-create-a-mobile-app/
- https://humanetech.com/

Making Computing Accessible


Making Computing Accessible
age 11
level 6
How can software be more accessible?
Phase 2: Product
age 17
level 12
How can computing help deal with an impairment?
Phase 4: Business

It's essential that everyone can benefit from computing and this Big Idea is about developing design skills for making computing accessible to those with disabilities or impairments. Key knowledge and skills in this Big Idea focus on understanding different types of disability or impairment and on developing an appreciation of how computing can help.

With a good understanding of how HCI technologies work, user-centric design skills and an appropriate attitude towards making computing accessible, students can then tackle authentic PBL challenges linked to UN SDG goals.

How can software be more accessible?

Domain – Human Computer Interaction

Idea – Making Computing Accessible

Level – 6

Time allocation – 10 hours

Learning Goals

Main

- Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users, CSTA 3A-AP-19
- Recommend improvements to design of computing experiences, MCSG

Supplementary

- Understand how screen readers and voice recognition programs work
- Make accessibility a regular part of design workflow
- Design for hearing, vision, mobility, and cognitive issues
- Consider accessibility in layout

Concepts

- When websites, technologies, or tools are poorly designed, they can create barriers that exclude people with disabilities
- Assistive technologies and design methods
- There is a very wide range of disabilities that need to be considered in software design

Challenges

Computational Thinking – Re-design a piece of software so it provides equal access and opportunity to people with a diverse range of hearing, movement, sight, and cognitive abilities

Data Literacy – use survey data to inform your re-design

Design Thinking – apply accessible technology design principles in your redesign

Supporting Content

Microsoft content

- **ML4S IDA**, Introduction to disability and accessibility, https://docs.microsoft.com/en-us/learn/modules/intro-accessibility-disability/
- **ML4S AF**, Accessibility fundamentals, https://docs.microsoft.com/en-us/learn/paths/accessibility-fundamentals/
- **ML4S DA**, Digital Accessibility, https://docs.microsoft.com/en-us/learn/modules/digital-accessibility/
- **ML4S CAC365**, Creating accessible content with Microsoft 365, https://docs.microsoft.com/en-us/learn/modules/accessible-content-m365/

How can computing help deal with an impairment?

Domain – Human Computer Interaction

Big Idea – Making Computing Accessible

Level – 12

SDG – 10. Reduced Inequalities

Time allocation – 6:40 hours

Learning Goals

Main

Factor-in accessibility when designing solutions, *Design Thinking, MCSG*

Supplementary

- Demonstrate knowledge of programming accessible web applications
- Critically assess the accessibility profile of software applications

Concepts

- Technology can play a significant role in helping people with impairments lead better lives
- Having people with impairments integrated into the modern workplace brings diversity benefits
- Open-source design frameworks such as Fluent Design System facilitate user-centred design

Challenges

Computational Thinking – Prototype a digital product that specifically addresses one significant impairment

Data Literacy – Gather and analyse data about the about the impairment that you are addressing

Design Thinking – design a way to help people with a specific impairment lead better lives

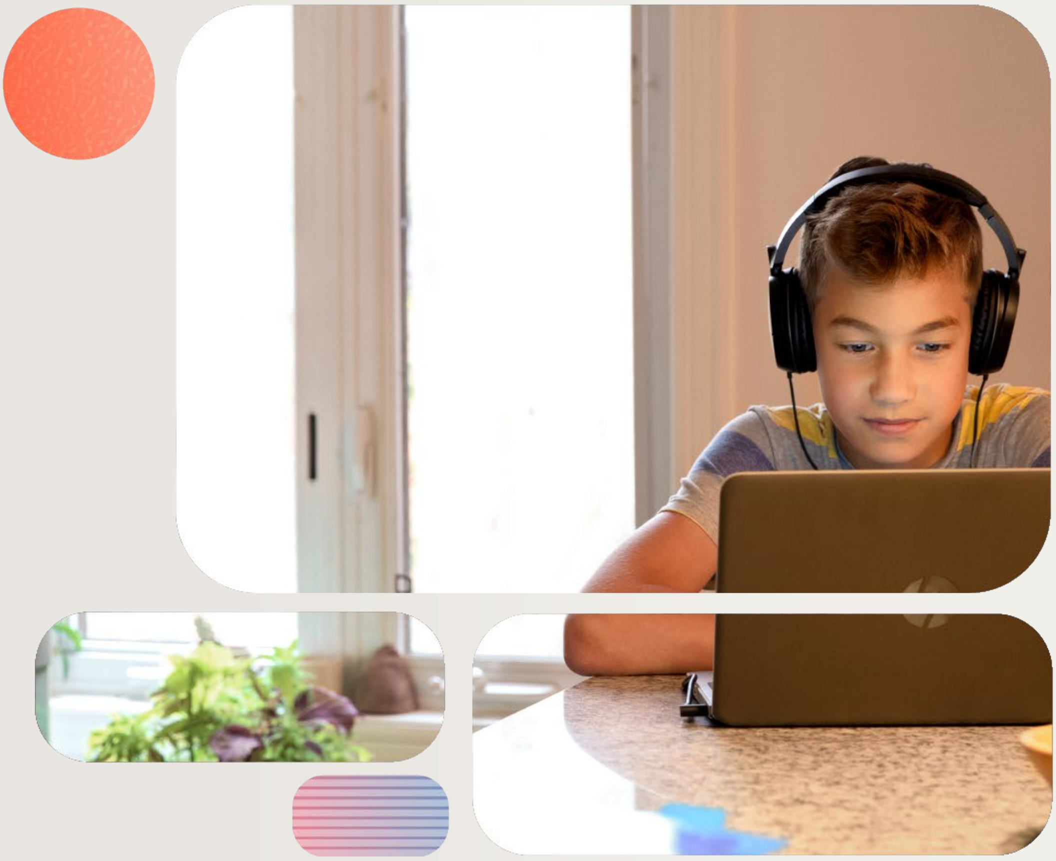
Supporting Content

Microsoft content

- SeeingAI – https://www.microsoft.com/en-us/ai/seeing-ai
- **MSA**, Microsoft Accessibility, https://www.microsoft.com/en-ca/accessibility/default.aspx
- **MID**, Microsoft Inclusive Design, https://www.microsoft.com/design/inclusive/
- Lynda UX, UX Accessibility, https://www.lynda.com/Accessibility-tutorials/Foundations-UX-Accessibility/435008-2.html
- Lynda AWD, Accessibility for Web Design, https://www.lynda.com/Web-Design-tutorials/Accessibility-Web-Design/606090-2.html

Supplementary content

- https://reasondigital.com/insights/four-ways-technology-can-help-disabled-people/
- https://www.bbc.co.uk/news/business-35427933

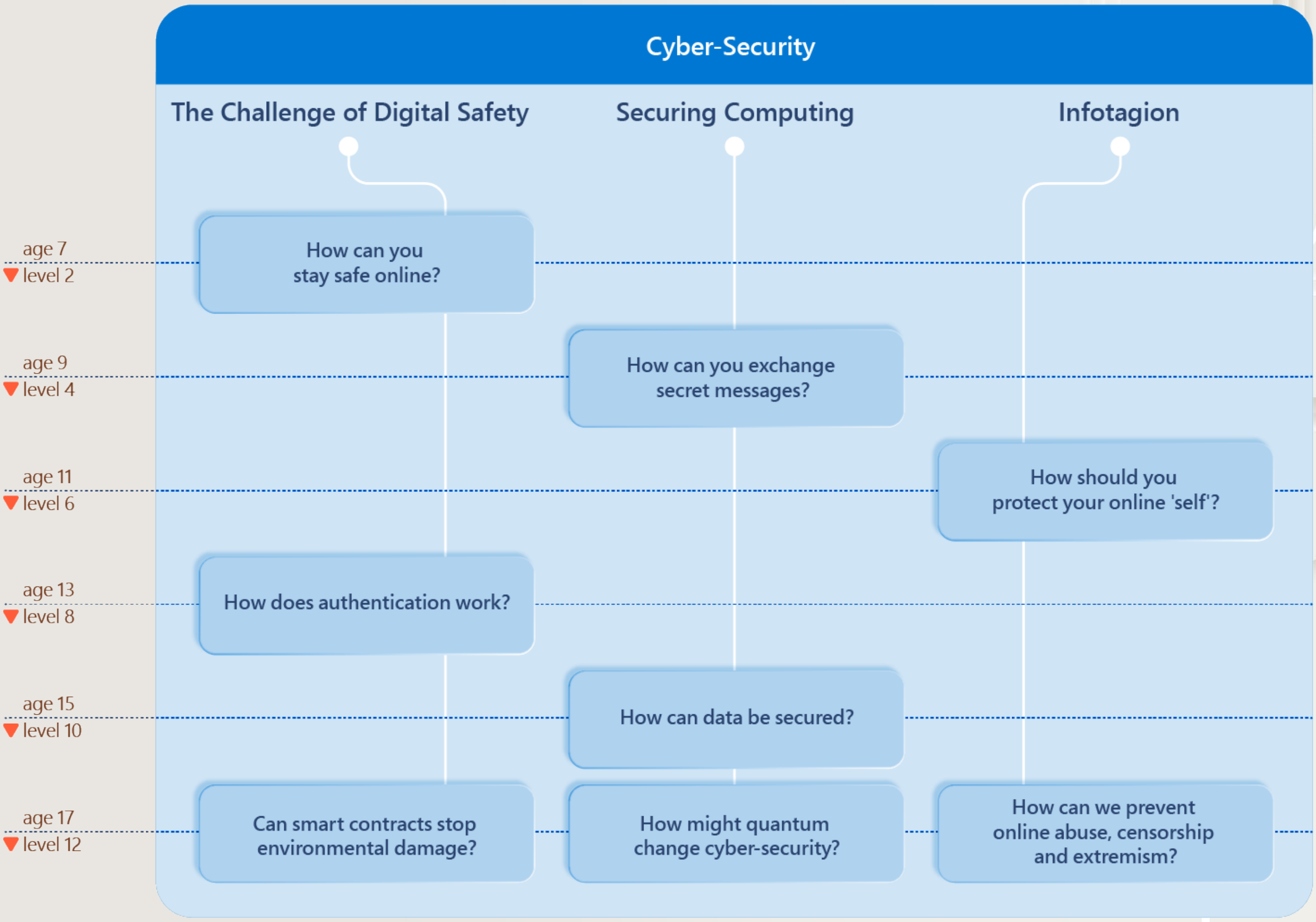


Cyber-Security

Cyber-crime earns criminals significantly more income than the combined revenues of Facebook, Amazon, Apple, Netflix, and Google.[lx] At the same time social media is driving misinformation and conspiracy theories, whilst the Dark Web enables extremists and organised crime – something that can be characterised as 'infotagion'. The combined economic and social damage caused by malicious internet use is vast and growing.[lxi]

The goal of this Domain is to develop a broad set of Cyber-Security skills by exploring the following Big Ideas:

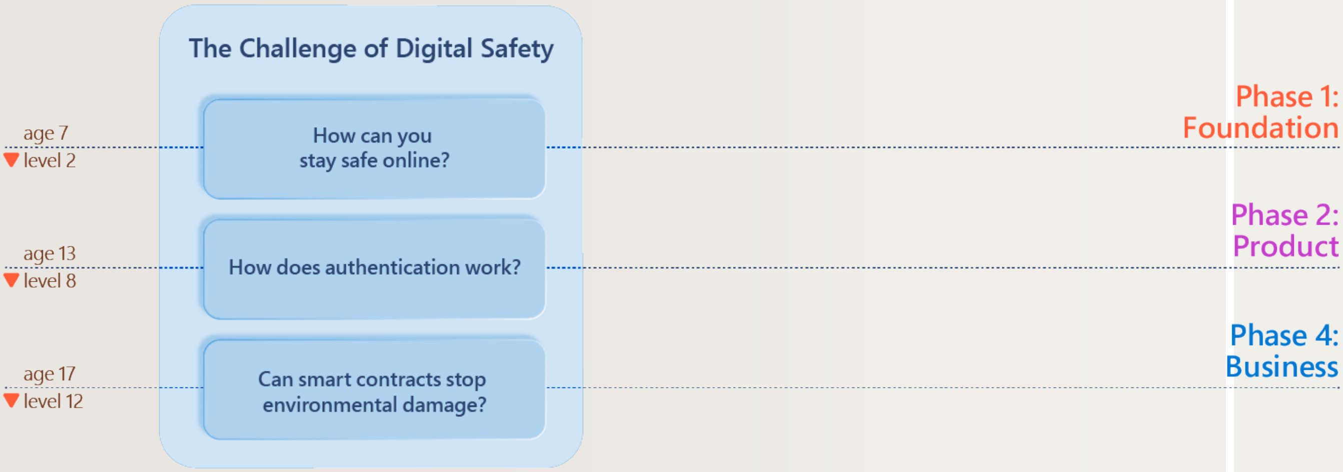
- The Challenge of Digital Safety
- Securing Computing
- Infotagion



Cyber-Security

	The Challenge of Digital Safety	Securing Computing	Infotagion
age 7 ▼ level 2	How can you stay safe online?		
age 9 ▼ level 4		How can you exchange secret messages?	
age 11 ▼ level 6			How should you protect your online 'self'?
age 13 ▼ level 8	How does authentication work?		
age 15 ▼ level 10		How can data be secured?	
age 17 ▼ level 12	Can smart contracts stop environmental damage?	How might quantum change cyber-security?	How can we prevent online abuse, censorship and extremism?

The Challenge of Digital Safety



	The Challenge of Digital Safety	
age 7 ▼ level 2	How can you stay safe online?	Phase 1: Foundation
age 13 ▼ level 8	How does authentication work?	Phase 2: Product
age 17 ▼ level 12	Can smart contracts stop environmental damage?	Phase 4: Business

An essential part of any curriculum today is teaching children how to stay safe online. This Big Idea builds on Internet safety to explore concepts such as trust, authentication, cyber-crime and smart contracts (Blockchain) through scientific and forensic lenses. For example, students will investigate how authentication works.

How can you stay safe online?

Domain – Cyber-Security

Big Idea – The Challenge of Digital Safety

Level – 1

Time allocation – 10 hours

Learning Goals

- Understand that personal data is collected, and that it is important to keep this information secure
- Develop respectful, responsible and safe behaviours with others online

Concepts

- The internet connects us to the world, and we need to be careful who we connect with online
- Personal and login information should not be shared with anyone we do not know
- A connected community needs to be respectful, responsible and safe for all

Challenges

Computational Thinking – Analyse and discuss online risks and behaviours

Data Literacy – Identify personal data types, data security

Design Thinking – Create a Family Tech Agreement

Supporting Content

Microsoft content

- **MDL LC4**, English Localised Course 4 – Participate safely and responsibly online, strong passwords, digital footprint, respect, responsible behaviour, https://www.microsoft.com/en-us/digital-literacy
- **CodWM OS**, Online Safety, https://education.minecraft.net/en-us/lessons/varnost-na-spletu

Supplementary content

- Code.org, Course 1, Lesson 17, Going Places Safely – https://stdio.code.org/s/course1/stage/17/puzzle/1
- UK Government (UKG), Teaching Online Safety in Schools – https://www.gov.uk/government/publications/teaching-online-safety-in-schools
- Australian Digital Technologies Hub, Communicating Online Safely – https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/communicating-safely-online

How does authentication work?

Domain – Cyber-Security

Big Idea – The Challenge of Digital Safety

Level – 8

Time allocation – 10 hours

Learning Goals

- Explain and use strong passwords to protect devices and information from unauthorized access, CSTA 1A-NI-04

- Discuss real-world cybers-security problems and how personal information can be protected, CSTA 1B-NI-05
- Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts, CSTA 3A-NI-06

Concepts

- Authentication verifies a user's identity to allow access to confidential data or systems
- There are a range of ways of authenticating identity, each with different merits
- Every security measure involves trade-offs between the accessibility and security of the system

Challenges

Computational Thinking – Test authentication factors for a web service

Data Literacy – Understand verification of data, cryptography keys, the concept of smart contracts

Design Thinking – Design the authentication sub-system for a given web service scenario

Supporting Content

Microsoft content

- **CodWM IinI**, The I in Identity, https://education.minecraft.net/en-us/lessons/the-i-in-identity
- **CodWM SCV**, Stranger in Crimwell Village, https://education.minecraft.net/en-us/lessons/stranger-in-crimwell-village
- **ML4S DIC**, Describe identity concepts, https://docs.microsoft.com/en-us/learn/modules/describe-identity-principles-concepts/
- **ML4S DSC**, Describe security and compliance concepts and methodologies, https://docs.microsoft.com/en-us/learn/modules/describe-security-concepts-methodologies/
- **ML4S PDiT**, Protect data in-transit and at rest, https://docs.microsoft.com/en-us/learn/modules/protect-data-transit-rest/
- **MS IP**, Microsoft Academic, Intellectual Property, https://academic.microsoft.com/topic/34974158/publication/search?q=Intellectual%20property&qe=And(Composite(F.FId%253D34974158)%252CTy%253D%270%27)&f=&orderBy=0

Can smart contracts stop environmental damage?

Domain – Cyber-Security

Idea – The Challenge of Digital Safety

Level – 12

SDG – 12. Responsible Production and Consumption

Time allocation – 6:40 hours

Learning Goals

Main

Design smart contract processes, *Design Thinking, MCFS*

Supplementary

- Demonstrate understanding of the fundamentals of blockchain technologies
- Build and deploy a basic smart contract on the blockchain

Concepts

- A blockchain is a distributed, decentralized public ledger
- Blockchain relies on public key cryptography to ensure that transactions are secure
- A smart-contract is a self-executing contract on the block-chain
- A smart contract is designed to promote contract adherence without the need for third party enforcement

Challenges

Computational Thinking – Write a smart contract using Azure Blockchain Workbench

Data Literacy – Understand elliptic curve digital signature algorithm (ECDSA)

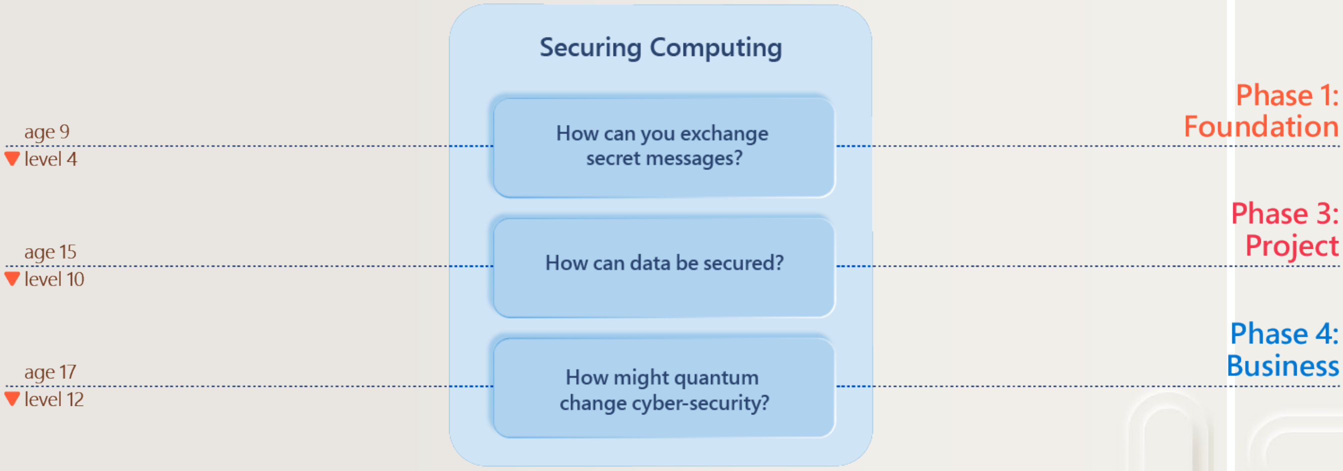
Design Thinking – Design a smart contract that will make forestry activities transparent to the public

Supporting Content

Microsoft content

- **BB**, Blockchain Basics, https://azure.microsoft.com/en-us/resources/videos/blockchain-basics/
- **ABS**, Azure Blockchain Service, https://docs.microsoft.com/en-us/azure/blockchain/service/send-transaction
- **BW**, Blockchain Workbench, https://docs.microsoft.com/bs-latn-ba/azure/blockchain/workbench/f

Securing Computing



With cyber-crime growing, it is extremely important for anyone studying computer science to understand the science of security, which includes concepts such as encryption and ciphers. A key goal of this Big Idea is to enable students to learn how to set up defences against cyber-attacks and secure sensitive and valuable information.

How can you exchange secret messages?

Domain – Securing Computing

Big Idea – Making Computing Safe

Level – 4

Time allocation – 10 hours

Learning Goals

Main

- Make a cipher work, MCSG
- Discuss real-world cyber-security problems and how personal information can be protected, CSTA 1B-NI-05

Supplementary

- Identify and uses codes and ciphers to exchange an encrypted message
- Explore patterns, frequencies and combinations to decrypt a secret message

Concepts

- A key is used to encrypt and decrypt information we want to keep private
- Information sent and received over the internet is broken down into smaller pieces called packets
- All information stored at the lowest level of the computer is in binary – off or on.

Challenges

Computational Thinking – Use Caesar Cipher algorithms to exchange secret messages

Data Literacy – Frequency analysis

Design Thinking – Design a code or cipher

Supporting Content

Microsoft content

- **MC CSP AP** 4,7, Computer Science Principles course, Unit 4 – Impact of Computing, Lesson 7, Cryptography, https://microsoft.github.io/makecode-csp/unit-4/impact-of-computing
- **MDL WWC** 4, Working With Computers, Keeping Your Computer Secure and Updated, https://www.linkedin.com/learning/working-with-computers-and-devices/working-with-computers-to-create-opportunity?autoAdvance=true&autoSkip=false&autoplay=true&resume=true

Supplementary content

- Computer Science Unplugged (CSU 4, 5, 6), Information Hiding, Cryptographic Protocols, Public Key Encryption – https://csunplugged.org/en/
- Code.org CSD Unit 4, including privacy, encryption, keys and passwords, key cryptography, cyber-crime, Simple Encryption – https://curriculum.code.org/csp-19/unit4/
- https://aca.edu.au/resources/cyber-crypto/
- https://groklearning.com/learn/cyber-78-py-crypto/crypto/27/

How can data be secured?

Domain – Cyber-Security

Big Idea – Securing Computing

Level – 10

SDG – 9. Industry, Innovation and Infrastructure

Time allocation – 10 hours

Learning Goals

Main

- Explain security issues that may lead to compromised systems, CSTA, 3B-AP-18
- Apply multiple methods of encryption to model the secure transmission of information, CSTA, 2-NI-06

Supplementary

Apply encryption techniques and understand the role of mathematics in encryption

Concepts

- Insecure software development practices pose serious security risks
- Encryption helps protects data you send, receive and store using a device
- Laws govern many aspects of computing, such as privacy, data, property, information and identity

Challenges

Computational Thinking – Research software coding errors: lack of bounds checking, poor input validation, circular references

Data Literacy – Prime Number Theory, Divisibility, Steganography

Design Thinking – Review case studies or events which present an ethical dilemma in data security

Supporting Content

Microsoft content

- **ML4S Sec1**, Microsoft Security, Compliance, and Identity Fundamentals: Describe the capabilities of Microsoft compliance solutions, https://docs.microsoft.com/en-us/learn/paths/describe-capabilities-of-microsoft-compliance-solutions/
- **ML4S Sec2**, Security, responsibility, and trust in Azure, https://docs.microsoft.com/en-gb/learn/modules/intro-to-security-in-azure/
- **ML4S Sec3**, Secure access to your applications by using Azure identity services, https://docs.microsoft.com/en-us/learn/modules/secure-access-azure-identity-services/
- **ML4S Sec4**, Explore the services and identity types of Azure AD, https://docs.microsoft.com/en-us/learn/modules/explore-basic-services-identity-types/
- **ML4S Sec5**, Describe the identity protection and governance capabilities of Azure AD, https://docs.microsoft.com/en-us/learn/modules/describe-identity-protection-governance-capabilities/
- **ML4S Sec6**, Secure your cloud resources with access control, https://docs.microsoft.com/en-us/learn/modules/cmu-secure-cloud-resources/
- **SYCD**, Secure your cloud data, https://docs.microsoft.com/en-us/learn/paths/secure-your-cloud-data/
- **PASTA**, Protect against security threats on Azure, https://docs.microsoft.com/en-us/learn/modules/protect-against-security-threats-azure/

- **FPS**, The Future of Privacy and Security, https://www.microsoft.com/en-us/research/video/opening-remarks-the-future-of-privacy-and-security/
- **INSA**, Implement network security in Azure -designed for administrator role, https://docs.microsoft.com/en-us/learn/paths/implement-network-security/
- **INSA**, Implement network security in Azure – designed for administrator role, https://docs.microsoft.com/en-us/learn/paths/implement-network-security/

Supplementary content

- Code.org CSD Unit 4, including privacy, encryption, keys and passwords, key cryptography, cyber-crime, Simple Encryption – https://curriculum.code.org/csp-19/unit4/
- https://aca.edu.au/resources/cyber-crypto/
- https://groklearning.com/learn/cyber-78-py-crypto/crypto/27/

How might quantum change cyber-security?

Domain – Cyber-Security

Idea – Securing Computing

Level – 12

SDG – 9. Industry, Innovation and Infrastructure

Time allocation – 6:40 hours

Learning Goals

Main

Explain the fundamentals of quantum computing, MCSG

Supplementary

- Demonstrate an understanding of cryptographic algorithms
- Write a basic program on a quantum computer

Concepts

- Quantum Supremacy is the goal of demonstrating that a quantum computer can solve a problem impossible for a classical computer
- Quantum computing is based on the Qubit rather than the Bit
- Shor's Algorithm shows that a quantum computer can find prime factors for large numbers in polynomial time
- Much of today's software including e-commerce is protected by cryptography that could be breached using a quantum computer
- Q# and Qiskit are examples of quantum programming languages that can be ran today on real and simulated quantum computers in the cloud
- RSA and ECDSA are examples of vulnerable algorithms in widespread use today

Challenges

Computational Thinking – Write a program to play Pong on a simulated quantum computer

Data Literacy – Calculate the time complexity of a program to find prime factors on a classical computer

Design Thinking – Design a plan to rollout post-quantum cryptography to e-commerce applications

Supporting Content

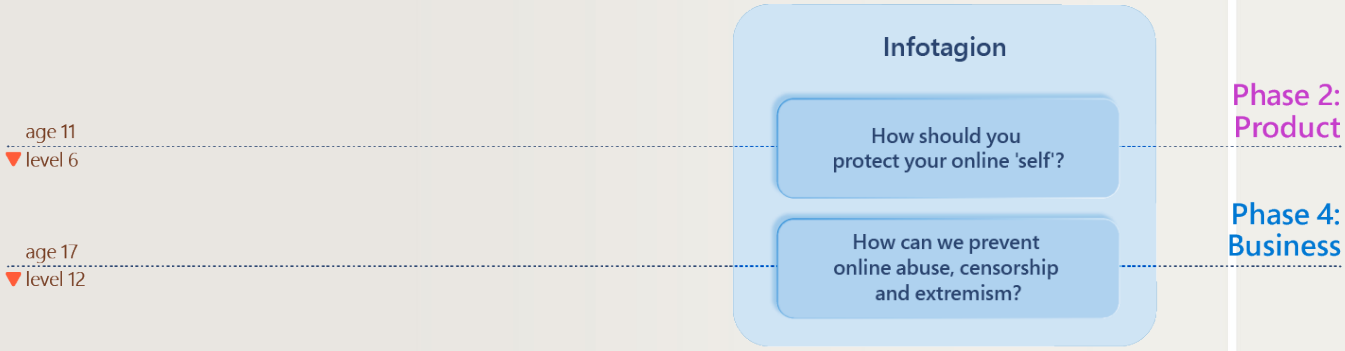
Microsoft content

- **ML4S Sec 7**, Quantum computing foundations, https://docs.microsoft.com/en-us/learn/paths/quantum-computing-fundamentals/
- **MSR SPC**, Security, Privacy and Cryptography, https://www.microsoft.com/en-us/research/research-area/security-privacy-cryptography/?facet%5Btax%5D%5Bmsr-research-area%5D%5B0%5D=13558&sort_by=most-recent

Supplementary content

- Shors Algorithm for Quantum: https://qiskit.org/textbook/ch-algorithms/shor.html
- Coding with Qiskit: https://www.youtube.com/playlist?list=PLOFEBzvs-Vvp2xg9-POLJhQwtVktIYGbY
- https://www.cryptomathic.com/news-events/blog/quantum-computing-and-its-impact-on-cryptography

Infotagion


Infotagion
age 11
level 6
How should you protect your online 'self'?
Phase 2: Product
age 17
level 12
How can we prevent online abuse, censorship and extremism?
Phase 4: Business

Whilst our computer systems are under attack from cyber-criminals, science itself is under attack from the misinformation industry which is often sponsored at state level. This Big Idea is about enabling students to understand the mechanics and business models underpinning disinformation and the damage it causes, equipping them with knowledge, skills and ethics to make the right choices as future developers.

With a good understanding of cyber-crime and how to defend against it, students will be equipped to specify and build secure services.

How should you protect your online 'self'?

Domain – Cyber-Security

Big Idea – Infotagion

Level – 4

Time allocation – 10 hours

Learning Goals

- Establish ethical protocols for the online world, MCSG
- Compare trade-offs associated with computing technologies that affect people's everyday activities and career options, CSTA 2-IC-20
- Describe trade-offs between allowing information to be public and keeping information private and secure, CSTA 2-IC-23

Concepts

- Security attacks often start with personal information that is publicly available online
- Data is encoded from a range of media to make predictions
- Using computing technologies has trade-offs, including privacy, communication and automation

Challenges

Computational Thinking – Audit your software and social media settings to analyse your online vulnerability

Data Literacy – Critically examine media reports of data breaches, trade-offs and cyber-risk

Design Thinking – Draw up a set of instructions to help people protect themselves online

Supporting Content

Microsoft content

- **ML4S SY**, Securing you: Basics and beyond, https://docs.microsoft.com/en-us/learn/modules/securing-you-basics-beyond/
- **MDL WCOL 2**, Working and Collaborating Online – WiFi, safety and responsible behaviour, video calls, file sharing, Lesson 2, https://www.linkedin.com/learning/working-and-collaborating-online/computing-with-confidence-getting-job-skills-and-staying-safe-online?autoAdvance=true&autoSkip=false&autoplay=true&resume=true

Supplementary content

- Code.org, Course 2, Lesson 18, Your Digital Footprint, https://studio.code.org/s/course2/stage/18/puzzle/1
- https://www.esafety.gov.au/educators/classroom-resources/making-good-choices-online
- https://groklearning.com/course/cyber-hs-infosec/

How can we prevent online abuse, censorship and extremism?

Domain – Cyber-Security

Big Idea – Infotagion

Level – 12

SDG – 16. Peace, Justice and Strong Institutions

Time allocation – 6:40 hours

Learning Goals

Main

Demonstrate an appreciation of technology law, MCSG

Supplementary

- Identify the motivations behind people who participate in online abuse and extremism
- Explain the theory and application of law to the cyber-realm

Concepts

- The search algorithms used by social media organisations impact the content viewed by the consumer
- Censorship legislation can be in direct conflict with a person's individual rights
- Responsibility for online safety falls to platform providers, users, government agencies but the zones of responsibility are not sharply demarcated
- A 'receiver operating characteristic' curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied

Challenges

Computational Thinking – Parameterize and run a machine learning program that will classify content as fake or not fake

Data Literacy – Analyse the performance of a fake news classification algorithm using a ROC curve

Design Thinking – Design a software feature that allows users to tag fake news, considering at which point (threshold) content should be removed

Supporting Content

Microsoft content

- **RI**, Responsible Innovation: The Next Wave of Design Thinking, https://medium.com/microsoft-design/responsible-innovation-the-next-wave-of-design-thinking-86bc9e9a8ae8
- **CodWM DC**, Digital Citizenship, https://education.minecraft.net/en-us/lessons/digital-citizenship-3

Supplementary content

- Infotagion: https://infotagion.com/
- The Center for Humane Technology, https://www.humanetech.com

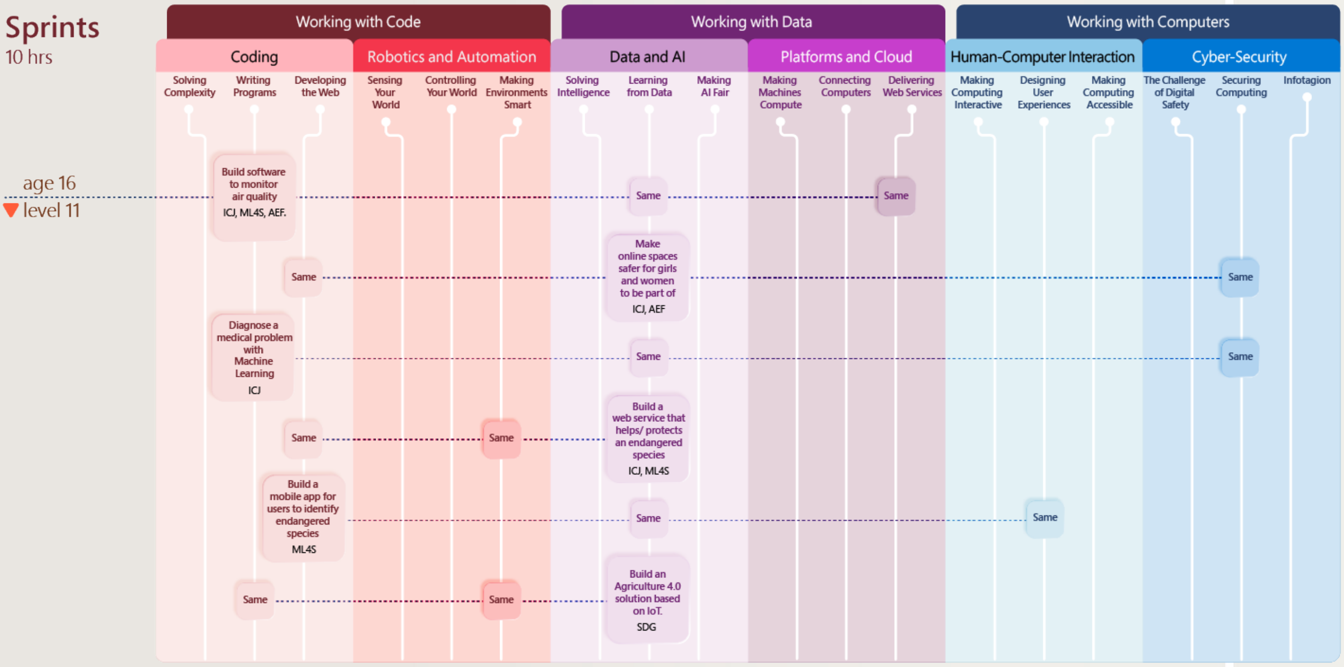
Project Phase Sprints

The Level 11 Sprints are intense 10-hour activities that combine taught content and short projects.

To teach the principle that CS solutions combine different disciplines, all of the Level 11 Sprints include content under the 'Writing Programmes' and 'Learning from Data' Domains, plus one other Domain so that by the end of the Sprints a range of combinations of disciplines will have been used.

Assessment criteria for each sprint –

- Produce use-case scenarios, plans sequences, goals, and outcomes
- Combine hardware, software and network technologies
- Modify, remix or incorporate parts of an existing program
- Apply Design Thinking – Empathise, Define, Big Ideate, Prototype, Test


Sprints
10 hrs
age 16
level 11
Working with Code
Working with Data
Working with Computers
Coding
Robotics and Automation
Data and AI
Platforms and Cloud
Human-Computer Interaction
Cyber-Security
Solving Complexity
Writing Programs
Developing the Web
Sensing Your World
Controlling Your World
Making Environments Smart
Solving Intelligence
Learning from Data
Making AI Fair
Making Machines Compute
Connecting Computers
Delivering Web Services
Making Computing Interactive
Designing User Experiences
Making Computing Accessible
The Challenge of Digital Safety
Securing Computing
Infotagion
Build software to monitor air quality ICJ, ML4S, AEF.
Same
Same
Same
Make online spaces safer for girls and women to be part of ICJ, AEF
Same
Diagnose a medical problem with Machine Learning ICJ
Same
Same
Same
Same
Build a web service that helps/ protects an endangered species ICJ, ML4S
Build a mobile app for users to identify endangered species ML4S
Same
Same
Same
Same
Build an Agriculture 4.0 solution based on IoT. SDG

Build software to monitor air quality

Domains – Coding, Data and AI, Platforms & Cloud

Big Ideas – Writing Programs, Learning From Data, Delivering Web Services

Level – 11

SDG – 13. Climate Action

Time allocation – 10 hours

Learning Goals

- Know what Azure Maps is, and how to create an an Azure Mpas account
- The different mapping options in Azure Maps
- How to show a map on a web page using the Azure Maps web SDK
- The GeoJSON format
- How to show a GeoJSON feature collection on a map

Challenges

Create a web page to show air quality data using Azure Maps

Supporting Content

Microsoft content

- **ML4S AQ**, Air quality, Maps, https://docs.microsoft.com/en-gb/learn/modules/azure-maps-track-air-pollution/

Make online spaces safer for girls and women to be part of

Domains – Coding, Data and AI, Human-Computer Interaction

Big Ideas – Writing Programs, Learning From Data, Securing Computing

Level – 11

SDG – 5. Achieve Gender Equality and Empower All women and Girls

Time allocation – 10 hours

Learning Goals

Develop skills in building secure services for a subset of the public

Challenges

Build a digital service for women or girls currently excluded from the digital world, for example make banking services available for women who are currently shut out of the banking system

Supporting Content

Microsoft content

- **AI4S**, AI for society, https://www.microsoft.com/en-us/ai/ai-lab-projects#primaryR5
- **GHAI**, Guidelines for Human-AI Interaction, https://www.microsoft.com/en-us/haxtoolkit/ai-guidelines/
- **FL**, Fairlearn, https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/
- **RML**, Responsible Machine Learning, https://docs.microsoft.com/en-us/azure/machine-learning/concept-responsible-ml
- **IncAI**, inclusive AI, https://www.microsoft.com/design/assets/inclusive/InclusiveDesign_InclusiveAI.pdf
- **MIAML**, Model interpretability in Azure Machine Learning, https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability?ocid=AID2463683&wt.mc_id=ai-c9-sejuare

Diagnose a medical problem with Machine Learning

Domains – Coding, Data and AI, Cyber-Security

Big Ideas – Writing Programs, Learning From Data, Securing Computing

Level – 11

SDG – 3. Good Health and Well Being

Time allocation – 10 hours

Learning Goals

- Understand the role of Computer Science in healthcare
- Develop Deep Learning and Neural Network skills
- Develop XAI skills
- Develop security skills
- Understand the need for balancing explainability with data security

Challenges

- Design a solution to diagnose a medical problem using deep learning and neural networks
- Choose an area where there is widely available public and anonymized data

Supporting Content

Microsoft content

- **ML4S PCDNN**, Make predictions from complex data with neural networks, https://docs.microsoft.com/en-gb/learn/modules/introduction-to-neural-networks/
- **KER**, How to Train Keras, https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-keras

Supplementary content

- Deep Mind at Moorfields Eye Hospital – https://www.moorfields.nhs.uk/content/latest-updates-deepmind-health
- NHS Predict breast cancer – https://breast.predict.nhs.uk

Build a web service that helps protects an endangered species

Domains – Coding, Data and AI, Platforms and Cloud

Big Ideas – Writing Programs, Learning From Data, Delivering Web Services

Level – 11

SDG – 15. Life on Land

Time allocation – 10 hours

Learning Goals

- Acquire the following toolset skills
- Azure Blob Storage
- Azure Functions Custom Vision Service,
- Microsoft Power BI
- Use a SQL Database

Challenges

Build a polar bear tracking solution that uploads images from a simulated array of cameras, analyses the images for polar bears, and visualise the output using Microsoft Power BI

Supporting Content

Microsoft content

- **ML4S TWPB**, Track wild polar bears, https://docs.microsoft.com/en-gb/learn/modules/build-ml-model-with-azure-stream-analytics/

Build a mobile app for users to identify endangered species

Domains – Coding, Data and AI, Human-Computer Interaction

Big Ideas – Writing Programs, Learning From Data, Designing User Experiences

Level – 11

SDG – 15. Life on Land

Time allocation – 10 hours

Learning Goals

- Develop the following skills:
- Building a fully functioning machine learning model
- Azure Cognitive Services
- Custom Vision
- UX design

Challenges

- Build a Machine Learning model to identify a bird in an image
- Design a user experience that is friendly to field ornithologists

Supporting Content

Microsoft content

- **ML4S CBCV**, Classify endangered bird species with Custom Vision, https://docs.microsoft.com/en-gb/learn/modules/cv-classify-bird-species/

Build an Agriculture 4.0 solution based on IoT

Domains – Coding, Data and AI, Platforms & Cloud

Big Ideas – Writing Programs, Making Environment Smarter, Learning From Data

Level – 11

SDG – 2. Zero Hunger

Time allocation – 10 hours

Learning Goals

- Understand the role of Computer Science in Agriculture 4.0
- Deepen Internet of Things skills
- Integrate a range of hardware and software solution components into a single solution

Challenges

Design and build an Urban or Vertical Farming solution

Supporting Content

Microsoft content

- **IoTIA**, IoT In Action – The Next Agricultural Revolution https://docs.microsoft.com/en-us/shows/internet-of-things-show/iot-in-action-the-next-agricultural-revolution

- **MF**, Manage farms, https://docs.microsoft.com/en-us/azure/industry/agriculture/manage-farms-in-azure-farmbeats

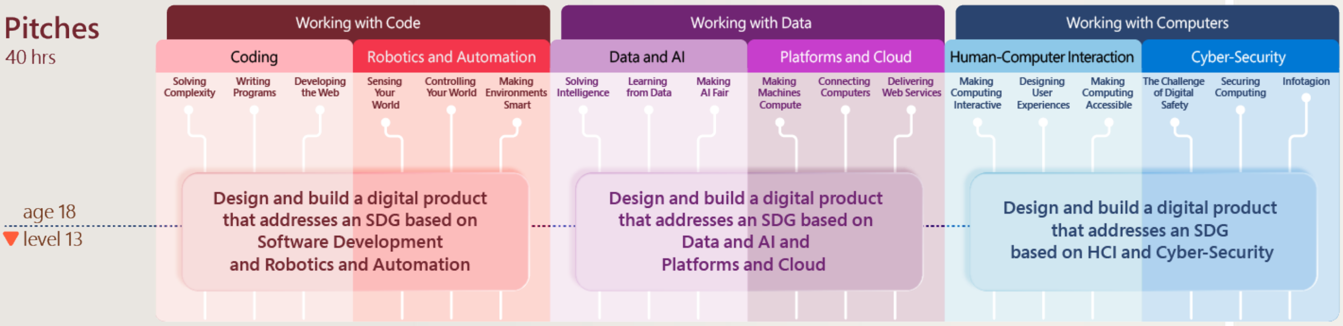
Supplementary content

- Vertical Farm Growing in the Basement of Baltic School – https://baltictriangle.co.uk/baltic-urban-farm/

Business Phase Pitches

The Level 13 Pitches are intense open-ended 40-hour projects that give students working in teams freedom to tackle any of the SDGs in which they have an interest. Each pitch requires students to combine two domains, so that each domain has been covered by the end of the academic year.

At the end of each Pitch, students will present their products to a 'board' of investors, which is likely to be made up of teachers, parents and local businesspeople.



Pitches
40 hrs

age 18
level 13

Working with Code

Coding: Solving Complexity; Writing Programs; Developing the Web

Robotics and Automation: Sensing Your World; Controlling Your World; Making Environments Smart

Design and build a digital product that addresses an SDG based on Software Development and Robotics and Automation

Working with Data

Data and AI: Solving Intelligence; Learning from Data; Making AI Fair

Platforms and Cloud: Making Machines Compute; Connecting Computers; Delivering Web Services

Design and build a digital product that addresses an SDG based on Data and AI and Platforms and Cloud

Working with Computers

Human-Computer Interaction: Making Computing Interactive; Designing User Experiences; Making Computing Accessible

Cyber-Security: The Challenge of Digital Safety; Securing Computing; Infotagion

Design and build a digital product that addresses an SDG based on HCI and Cyber-Security

Each of the Pitches has the following characteristics

Level – 13

SDG – Any

Time allocation – 40 hours

Learning Goals

- Further develop project management, team-working and solution development skills
- Develop entrepreneurial skills and a sense of commercial realism as a context for Computer Science
- Develop presentation skills

Challenges

- Design and build a digital product that addresses an SDG based on Coding and Robotics and Automation.
- Convince a 'board of investors' that the Product is worth investing in

Supporting Content

- Business Model Canvas
- Hackathons

Marking criteria for each Pitch

- Apply sound fundamentals of system organisation, architecture and solution design
- Collaborate effectively in a software team using version control and continuous deployment for commerce-ready software
- Effectively use prototyping
- Pitch convincingly to investors
- Students are required to design and build a digital product that addresses an SDG based on:
- 'Coding' and 'Robotics and Automation'
- 'Data and AI' and 'Platforms and Cloud'
- 'Human-Computer Interaction' and 'Cyber-Security'

Content Used in the MCSG

Make Code	Cue micro:bit – Toys; Banana Keyboard *Computer Science Intro:* Variables, Sprites, Motion, Loops, User input, JavaScript, Namespaces, functions, Info and Button Press Events, Array Manipulation Serial data Bluetooth pairing
Minecraft Education	*Computing with Minecraft:* • Describe what coding is. Code an Agent to move in different directions. Use on chat commands. • Create multiple lines of code; Merge multiple types of coding features; Use block code and/or Java Script as a coding language. • Use loops to code more efficiently, calculate in-game actions • Understand that code is determined by design goals • Use code to design a moving system. • Use multiple moving parts to create an animated feature. *Coding with Minecraft:* • Coordinates • Variables • Conditionals
Hacking STEM	• Lemon battery and switch • Analyzing the astronauts’ photos of Earth to predict climate change • How are ocean currents formed? Salinity and Temp sensors • Harnessing Electricity to Communicate • Building Machines That Emulate Humans. • Analysing Windspeed With Anemometers • Farmbeats
Digi Girlz	• Alice Envisions the Future: • Achieve gender equality and empower all women and girls • Reduced Inequalities • Good Health and Well-Being • Climate Action
Imagine Cup Junior	• Preparing data for Machine Learning; • Intro to Cloud Computing; Introduction to AI; API, Public Datasets; Introduction to Python. • Machine Learning. Datasets. Supervised, Unsupervised. Clustering. BOTs. • Applying AI. Facial recognition and sentiment analysis. NLP. PowerBI. • Deep Learning and Neural Networks. • AI for Good. AI for Earth, Fishing, Agriculture, Heritage, Humanitarian

Microsoft Digital Literacy Course	- Work with Computers - Access information online - Communicate online - Participate safely and responsibly online - Create digital content - Collaborate and manage content digitally
Microsoft Learn for Students (ML4S)	- Create an Azure account - Manager services with the Azure portal - Import standard library modules - Build a simple website using HTML, CSS, and JavaScript - Introduction to Neural Networks - Write your first C# Code - Principles of cloud computing - Security, responsibility, and trust in Azure - Air quality maps - Foundation of data science - Make predictions from complex data with neural networks - Build an AI web app by using Python and Flask - Architecture and service guarantees - Networking options - Track wild polar bears - Create Mobil App with Xmarin - Introduction to Azure IoT - Introduction to Bash - Computer options - Apply and monitor infrastructure standards with Azure Policy - Classify endangered bird species with Custom Vision - Use visuals in Power BI - Analyze climate data with Azure Notebooks - Storage options - Weather data alert - Python Object Oriented Programming - Classify images with the Custom Vision - First-time Python
Microsoft content for professionals	- Fundamentals of Computer Networking - MS Power Automate - GitHub Student Developer Pack - Introduction to Robotics - Responsible AI Resources - Fluent - SPC Security, Privacy and Cryptography - Power BI 1 - Fundamentals of Computer Networking - Azure Create a Web App - Microsoft CityNex

	• Responsible Machine Learning • Azure Blockchain Service • Post Quantum Cryptography • Responsible innovation: The Next Wave of Design Thinking • QuickStart: Create a virtual network using the Azure portal • Robot Challenge Lab • Azure IoT Developer Kit • Intro to Deep Learning • Blockchain Basics • Python for Beginners • Microsoft Accessibility • Introduction to Quantum Computing • Azure in Autonomous Driving • Introduction to Azure IoTn-to-azure-iot/ • How to Train Keras • Blockchain Workbench • Get started with Mixed Reality • Microsoft Inclusive Design • Post Quantum Cryptography • AI Robot Lab • Deep Learning Versus Machine Learning • Neural Network Regression • Getting Started with the Quantum Development Kit
Lynda	• Accessibility for Web Design • Visio 2019 Essential Training • Agile Foundations • UX Accessibility • Agile Project Management with Microsoft Project
Code.org	• Course 1, Lesson 17, Going Places Safely • Course 2, Lesson 18, Your Digital Footprint • Problem-Solving. What is a computer? IO, processes, and storage • Web Development. HTML, CSS, build a personal portfolio website • The Design Process. User-Centered Design, UX, prototyping, testing • Physical Computing. Fitting coding to screens and using virtual sensor data. • CS Principles, Big Data and Privacy, Unit 4, Lesson 7. Simple Encryption including Caesar Cipher
CS Unplugged	• Binary Numbers • Image Representation • Information Hiding • Cryptographic Protocols • Public Key Encryption • Kidbots • The Turing Test • Class Simulation of a Computer

Intel "Journey Inside"	1. Introduction to Computers. History of a computer. Four components of a computer. Compare computer "brains" with the human brain. 2. Circuits and switches 3. Digital Information 4. Microprocessors 5. The Internet 6. Technology and Society
Learn-tech	• How to Make a Mind • XAI • SIR Model – Covid-19 Demystified
CS4FN	• Machine Learning Board Game • Programming php
NASA	NASA Software Safety Guidebook
CS Field Guide NZ	Human-Computer Interaction
The Barbican	AI: More Than Human Virtual Exhibition
UK Gov	UK Government, Teaching Online Safety in Schools
Barefoot	Shopping list, Story sequencing, Dance moves, Pizza party, House patterns, Building blocks, Creating patterns, Barefoot Zoo, Sorting it out, Colorful Kits

MCSG Developers

The MCSG was developed by a group of current and former educators, computing professionals and researchers, as well as Microsoft employees currently working in the education space around the world. Advisors to the project included **ScienceScope,** an award-winning schools technology company that originated data-logging in schools, and were a key partner of the BBC in delivering the micro:bit project worldwide.

Team

Mike Lloyd – Project Leader
Technology leader Mike Lloyd's passion is to democratise Deep Tech. In September 2013 Mike founded learn-tech.io, a technology start-up, to build learning solutions focussed on AI, Internet of Things (IoT), and climate change mitigation. learn-tech.io has customers for its products and services in Europe and Asia, and Mike works with a wide range of organisations including Pearson, ScienceScope, Microsoft, Queen Mary University of London, Singapore University of Social Sciences, Bankers' Lab and Barclays Bank. Mike is the author of "Schooling at the Speed of Thought", a guide to transforming education, and numerous courses including "AI Demystified", "AI for Leaders" and "Digital Technology Demystified", now published by Pearson. He is also the inventor of "ZEP Island", an educational game that helps people learn how technology, data and IoT can be used to mitigate against climate change.

Support team

Dr. Kristen Weatherby is a researcher and education consultant helping start-ups, SMEs and other organisations understand, measure, and communicate their impact in education. Most recently, she helped develop and launch University College London's EDUCATE accelerator supporting edtech start-ups in their development of world class products for education. At the OECD, Dr. Weatherby led the 2013 cycle of the Teaching and Learning International Survey (TALIS), working with governments and partners to survey 104,000 teachers in 34 countries about their working conditions and learning environments. At Microsoft, she led initiatives to support schools and teachers integrating technology into teaching in the United States, the United Kingdom and worldwide. She has a PhD in education technology from the UCL Institute of Education, a Master's in teaching from the University of Michigan and is a governor of her local primary school.

John Curry is an AI practitioner, solution architect and educator who provides AI based solutions and services to clients across all sectors. He made the transition from solution architect to data scientist in the early 2010's culminating in a Master's in Business Analytics from University College Dublin in 2013. Most recently he has been working with SoptAI of Singapore to develop state-of-the-art demand forecasting solutions for businesses in heavy industry, supply chain and FMCG. He has also partnered with learn-tech-io to develop a framework Computer Science K12 curricula for Microsoft. John has become increasingly involved in the development of AI talent. During his tenure as Product Manager for DataSpark, John founded the internship program in 2014, sourcing talented Computer Science students from 10 countries to work on geospatial analytics. Since 2018, he has been designing and delivering courses on Fintech and Data Science to the Institute of Banking in Ireland. In 2019, John founded Last Mile Artificial Intelligence to help businesses bridge the last mile gap to successful application of AI to practical problems and serves clients in Ireland, the UK, Singapore and the USA.

Donna Buckley is a mathematics teacher at John Curtin College of the Arts, West Australia's only selective school for gifted and talented students in the Arts. The college is a Teacher Development School in STEM and Donna develops resources to support teachers across the state with their interdisciplinary approaches to their STEM learning programs. Specialising in mathematics, cyber-security, cryptography and the integration of programming across the curriculum she has presented on these topics at Mathematics, STEM and Cyber Conferences across Australia. She is a member of the Maths Association of Western Australia Board, coordinating Maths Talent Quest, a student activity that encourages students to consider real world problems and apply mathematical and statistical thinking processes to solve them. Volunteering as a tutor at the Perth Girls Programming Network, she supports girls in secondary schools to develop their Python programming skills. A member of the Day of STEM Education Advisory Board, she provides feedback on their programs and actively promotes their STEM careers resources in Cyber-security and Data Science. Donna is an author of the MAWA Mathematics Essential Textbooks Units 1 – 4.

Appendix

Glossary

AI – Artificial Intelligence.

API – Application Programming Interface.

App – Application, usually refers to programs that run on mobile phones.

ASCD, Association for Supervision and Curriculum Development – an organisation that empowers educators to achieve excellence in learning, teaching, and leading so that every child is healthy, safe, engaged, supported, and challenged.

Azure – Microsoft's set of Cloud services.

BI, Business Intelligence – strategies and technologies used by enterprises for collecting and analysing data and sharing of business information.

Block-based programming – as opposed to text-based programming, block-based programming refers to programming languages and environments that separates executable actions into modular portions called blocks. Block-based programming allows a 'drag and drop' approach to programming and is widely seen as a stepping-stone to students learning text-based programming.

Blockchain – a decentralised distributed ledger designed to solve the double spend problem associated with digital money, but also applied to Smart Contract scenarios.

Bot – Software application that runs automated tasks over the Internet.

CAS, 'Computing At School' – a UK community of teachers, academics and industry professionals that provides access to resources.

Chatbot – a software application used to conduct an online chat conversation via text or text-to-speech in lieu of providing direct conversation with a live human agent.

Cloud – on demand availability of computer systems resources, especially data storage, computing power and applications.

Computing with Minecraft, Minecraft Education – a version of the Microsoft games software, Minecraft designed to help students learn to code and apply coding across STEM subjects.

Computer Science (CS) – the theory and practice of computation, algorithm development, computational problems, and the design of computer systems, including hardware, software and applications.

Computer Science Teachers Association (CSTA) – a professional association whose mission is to "empower, engage and advocate for K-12 CS teachers worldwide".

CSU Computer Science Unplugged – a primary school Computer Science program that teaches the principles of Computer Science through playground and classroom games without using computers. The program is carried out by the Computer Science Education Research Group, University of Canterbury, New Zealand.

Cyber-Security – the application of technologies, processes and controls to protect systems, networks, programs, devices and data from cyber-attacks.

Data Science (DS) – an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data. DS involves model-building, system design and implementation, statistical analysis and data modelling.

Dry OO – "Don't Repeat Yourself" is a principle of software development aimed at reducing repetition of software patterns, replacing it with abstractions or using data normalization to avoid redundancy.

ECDSA, Elliptic Curve Digital Signature Algorithm – a Cryptographic algorithm used by Bitcoin to ensure that funds can only be spent by their rightful owners. Concepts related to ECDSA include Public and Private Keys.

FMCG – 'fast moving consumer goods' refer to products that are sold at a low cost and relatively fast. These goods are also referred to as consumer packaged goods (abbreviated CPG). Examples of FMCG include non-durable products, such as toiletries, over the counter medications, and packaged foods.

GEOJSON – an open standard geospatial data interchange format based on JavaScript Object Notation (JSON). It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties, and their spatial extents. It represents simple geographical features, along with their non-spatial attributes.

Git – a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

GPIO – General-purpose input/output is a digital signal pin on an integrated circuit or electronic circuit board whose behaviour is controllable by the user at run time.

HCI, Human-Computer Interaction – a multidisciplinary field of study focusing on the design of the interaction between humans (the users) and computers. While initially concerned with computers, HCI has since expanded to cover almost all forms of information technology design.

IB, International Baccalaureate – an international curriculum which offers an education for students from age 3 to 19, comprising of four programmes that focus on teaching students to think critically and independently, and how to inquire with care and logic.

IC, Integrated Circuit, – a set of electronic circuits on one small flat piece (or "chip") of semiconductor material that is normally silicon.

ICT, Information and Communications Technology – an extensional term for information technology IT that stresses the role of telecommunications and computers. ICT is a widely used term to describe computing education in schools.

IoT, Internet of Things – a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. The definition of the Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things.

ISCED 2011, International Standard Classification of Education – the International Standard Classification of Education (ISCED 2011) provides a comprehensive framework for organising education programmes and qualification by applying uniform and internationally agreed definitions to facilitate comparisons of education systems across countries.

JSON JavaScript Object Notation – a language often used when data is sent between a server and a web page.

LAN, Local Area Network – a LAN is a computer network that typically spans single room, building or group of buildings. LANs can also be connected to other LANs to form WANs (Wide Area Networks).

Learning Goal – brief statements that describe what students will be expected to learn by the end of school year, course, unit, lesson, project, or class period.

LIME Library – "Local Interpretable Model-agnostic Explanations" is used to explain predictions in AI systems.

LoRa, Long Range – is a low-power WAN (LPWAN) protocol.

MDL Microsoft Digital Literacy – a course designed for anyone with basic reading skills who wants to learn the fundamentals of using digital technologies.

MoE – Ministry of Education

MS4S, Microsoft Learn for Students – a completely free, open training platform available to anyone who has an interest to learn about Microsoft products.

MR – Mixed Reality is the result of blending the physical world with the digital world and includes and includes technologies such as Augmented and Virtual Realities.

NLP Natural Language Processing – a method for programming computers to process and analyse large amounts of natural language data.

OECD, Organization for Economic Cooperation and Development – an international organisation that works to build better policies for better lives.

OOP Object-Oriented Programming – a programming methodology based on objects, instead of just functions and procedures. These objects are organized into classes, which allow individual objects to be grouped together.

OpenIDEO – an open innovation model that allows the organization to help people worldwide break barriers, find support, and iterate on the ideas of many to create real change.

PBL, Problem-Based Learning – a teaching method in which complex real-world problems are used as the vehicle to promote student learning of concepts and principles as opposed to direct presentation of facts and concepts.

PfB, Python for Beginners – resources for learning how to code with Python.

Process patterns – can be defined as the set of activities, actions, work tasks or work products and similar related behaviour followed in a software development life cycle.

Qiskit – a platform released by IBM to enable users to more easily write code and run experiments on the quantum processor and simulator.

Quantum Computing – the use of quantum-mechanical phenomena such as superposition and entanglement to perform computation. Quantum computers are believed to be able to solve certain computational problems, such as integer factorisation, substantially faster than classical computers.

QuBit – In quantum computing, a qubit is the basic unit of quantum information – the quantum version of the classical binary bit.

RMSE, Root Mean Square Error – the measure of the differences between values predicted by a model or an estimator and the values observed.

ROC curve – Receiver Operating Characteristic curve – is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

RPA, Robotic Process Automation – a form of business process automation technology based on software robots (bots) or on artificial intelligence (AI)/digital workers.

RSA (algorithm) – RSA (Rivest–Shamir–Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission.

SAAS Software as a Service – a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. It is sometimes referred to as "on-demand software".

SDG, Sustainable Development Goals – a blueprint of 17 interconnected goals devised and promoted by the UN to achieve a better and more sustainable future for all. They address the global challenges we face, including those related to poverty, inequality, climate change, environmental degradation, peace and justice.

SME – Small and Medium Enterprises

SQL – Structured Query Language is a domain-specific language used in programming and designed for managing data held in a relational database management system. It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.

SW – Software

Time complexity – the amount of time it takes to run an algorithm.

UBD Understanding by Design – a framework which offers a planning process and structure to guide curriculum, assessment, and instruction.

UK Exam board – OCR, Oxford, Cambridge, RSA, AQA are independent organisations that set and process examinations in schools.

UML – The Unified Modelling Language (UML) is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

UX – a user's experience of using a piece of technology.

WAN, Wide Area Network – a system of connected Local Area Networks (LANs) typically spanning a relatively large geographical area.

XAI – Explainable AI (XAI) refers to methods and techniques in the application of artificial intelligence technology (AI) such that the results of the solution can be understood by human experts. It contrasts with the concept of the "black box" in machine learning where even their designers cannot explain why the AI arrived at a specific decision. XAI is an implementation of the social right to explanation.

Influences

- Successful Computer Science **inclusion programmes**, e.g. DigiGirlz
- **Code.org**
- **Simon Payton-Jones**
- **London Interdisciplinary School**. Today's tough problems do not map into the traditional subjects around which most education is organised.[lxii] The London Interdisciplinary School (LIS), which will launch in 2021, will scrap traditional academic subjects and offer a degree designed to tackle real-world issues. The curriculum is built around interdisciplinary problems – knife crime, childhood obesity, palm oil in supply chains, plastic pollution – as well as quantitative and qualitative research skills. Here, the problem, not the subject, sits at the centre of the curriculum. Students need to develop skills that enable them to thrive in an AI-infused, automated world: e.g. collaboration between people and machines, critical thinking, and data management.
- **Innovation and Entrepreneurship courses** – e.g. University of Bristol, Imperial College, Queen Mary University London
- **Tech incubators**, e.g. Y Combinator and SETsquared. Given that an end goal of the MCSG is to enable graduating students to build commercial computing solutions to real-world problem, the combined disciplines of technological creativity and entrepreneurialism fostered in tech incubators provide useful guidance. Y Combinator has incubated Airbnb, DropBox, Reddit, and Docker, whilst SETsquared, where `learn-tech.io` was incubated, was voted the world's top incubator in 2015 and 2018.
- **Deep Science Ventures** takes the incubation approach a step further by clustering businesses that they invest in around problem domains
- **Deep Mind** – the company responsible for major breakthroughs in AI – focuses on the single problem Domain of "solving intelligence".
- **Open IDEO** – is an open innovation platform where people from all corners of the world collaboratively tackle some of the toughest global issues. It can be regarded as a hybrid between Crowdsourcing and Open Design.
- **UN Sustainable Development Goals (SDG)**
- **Finland**, which is preparing to move from a "teaching by subject" to a "teaching by topic" approach in its secondary schools
    - **Computing Curricula 2020**, https://cc2020.nsparc.msstate.edu
    - **OECD**, http://www.oecd.org/education/2030-project/

References

[i] https://www.accenture.com/_acnmedia/pdf-63/accenture-new-skills-now-inclusion-in-the-digital.pdf *and* https://news.sky.com/video/coronavirus-the-outbreak-will-accelerate-digital-transformation-11960343

[ii] https://www.bbc.co.uk/news/av/stories-52962878/the-teenage-inventor-on-a-mission

[iii] https://www.bbc.co.uk/news/av/stories-50988283/teaching-school-kids-to-be-entrepreneurs

[iv] The event, *Hacking future skills: Computer Science Education*, was held online on 25 June 2020.

[v] See Dochy et al, 2003, Merritt et al, 2017, and Holmes, V. & Hwang, Y. (2016). Exploring the effects of project-based learning in secondary mathematics education. *The Journal of Education Research*, 109(5), 449-463.

[vi] Dochy et al, 2003.

[vii] Holmes & Hwang, 2016.

[viii] Merritt et al, 2017.

[ix] Ibid.

[x] Ashcraft, C., Eger, E., Friend, M. (2012). *Girls in IT: The facts*. National Center for Women and Information Technology. Accessed from: https://www.ncwit.org/sites/default/files/resources/girlsinit_report2012_final.pdf

[xi] Weatherby, K. (2017), *Teacher participation in online communities of practice: a mixed-methods study of community, context and practice*, University College London, London. Accessed from: https://discovery.ucl.ac.uk/id/eprint/1566655/1/Weatherby_2017%20K%20Weatherby%20PhD%20_-%20FINAL.pdf

[xii] https://www.un.org/development/desa/disabilities/envision2030.html

[xiii] https://www.pearson.com/uk/educators/schools/news/schools-blog/2022/02/abacus-and-low-stakes-assessment.html

[xiv] https://www.pearsonvue.co.uk/Documents/Grow-your-programme/VoC.aspx

[xv] https://learn.microsoft.com/en-us/certifications/college-credit

[xvi] https://hbr.org/2019/11/6-reasons-why-higher-education-needs-to-be-disrupted

[xvii] https://dera.ioe.ac.uk/5284/1/finalreport.pdf

[xviii] https://www.oecd.org/cfe/leed/BGP_Entrepreneurship-in-Education.pdf

[xix] Lloyd, Mike (2010), Schooling at the Speed of Thought, Spiderwise

[xx] https://www.skillstruck.com/blog/5-ways-to-improve-equity-in-computer-science

[xxi] Ben Francis, founder of online clothing apparel company Gymshark, https://www.bbc.co.uk/news/technology-62098767

[xxii] https://www.edsurge.com/news/2020-02-24-4-critical-characteristics-of-high-quality-instructional-content

[xxiii] https://www.bbc.co.uk/bitesize/guides/z4rbcj6/revision/1

[xxiv] Skinner, E. A., & Belmont, M. J. (1993). Motivation in the Classroom: Reciprocal Effects of Teacher Behavior and Student Engagement across the School Year. *Journal of Educational Psychology*

[xxv] https://www.ted.com/talks/conrad_wolfram_teaching_kids_real_math_with_computers?language=en

[xxvi] Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). *How people learn* (Vol. 11). Washington, DC: National Academy Press.

[xxvii] ibid

[xxviii] Wiggins, G., & McTighe, J. (2005). *Understanding by design*. Alexandria, VA: Association for Supervision and Curriculum Development.

[xxix] Darling-Hammond, L., & Bransford, J. (Eds.). (2007). *Preparing teachers for a changing world: What teachers should learn and be able to do*. San Francisco, CA: John Wiley & Sons.

[xxx] ibid

[xxxi] Bransford et al, 2000

[xxxii] http://uis.unesco.org/sites/default/files/documents/international-standard-classification-of-education-isced-2011-en.pdf

[xxxiii] https://www.researchgate.net/publication/283550497_Design_Thinking_pedagogy_the_Educational_Design_Ladder

[xxxiv] There is no http://www.mempowered.com/children/international-curricula

[xxxv] https://en.wikipedia.org/wiki/Academic_term

[xxxvi] https://www.computingatschool.org.uk/

[xxxvii] https://k12cs.org

[xxxviii] https://oecdedutoday.com/should-schools-teach-coding/

[xxxix] https://static1.squarespace.com/static/57c6b79629687fde090a0fdd/t/5b19b2f2aa4a99e99b26b6bb/1528410876119/dschool_bootleg_deck_2018_final_sm+%282%29.pdf

[xl] https://www.tes.com/teaching-resource/computational-thinking-guide-and-progression-pathways-6436712

[xli] https://www.esd105.org/cms/lib/WA01920102/Centricity/Domain/175/2017%20CSTA%20K-12%20Standards%20Progression%20Chart%20CT.pdf

[xlii] https://csteachers.org/Page/standards

[xliii] https://csteachers.org/Page/standards

[xliv] https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/

[xlv] https://www.australiancurriculum.edu.au/resources/national-literacy-and-numeracy-learning-progressions/

[xlvi] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf

[xlvii] https://educationblog.microsoft.com/en-us/2019/12/empowering-students-to-be-future-ready/

[xlviii] Pearson Vue, Certiport white paper – *Ten reasons schools should offer Microsoft certification to students*

[xlix] https://codingcredentials.com

[l] https://codingcredentials.com/coding-in-minecraft-exam/

[li] https://classic.csunplugged.org/activities/

[lii] https://www.tiobe.com/tiobe-index/

[liii] https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP

[liv] https://www.codingninjas.com/blog/2021/07/29/c-vs-java-vs-python-which-one-to-choose/

[lv] https://blogs.ei.columbia.edu/2019/09/23/climate-change-impacts-water/

[lvi] https://www.nature.com/articles/d41586-018-05849-9

[lvii] https://blogs.ei.columbia.edu/2019/09/23/climate-change-impacts-water/

[lviii] https://www.statista.com/statistics/740440/worldwide-robotic-process-automation-market-size/

[lix] https://royalsociety.org/-/media/policy/projects/ai-and-work/frontier-review-the-impact-of-AI-on-work.pdf

[lx] https://www.experian.com/blogs/ask-experian/cybercrime-the-1-5-trillion-problem/

[lxi] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/60943/the-cost-of-cyber-crime-full-report.pdf

[lxii] *Forces Shaping the U.S.* Academic Engineering Research Enterprise, By National Academy of Engineering, Committee on Academic Engineering Research

Other Useful References

Process Patterns – https://study.com/academy/lesson/process-patterns-in-software-engineering-types-examples.html

Curriculum Structure – https://www.australiancurriculum.edu.au/f-10-curriculum/structure/

International CS Education - https://docs.google.com/document/d/1H171Mu2RKzD9Qvp38sjKu1vuXX524XqdHtUf0BFzpWI/pub

Microsoft