About this document

The aim of this document is to support teachers in primary schools (K5) to harness Universal Design for Learning in the teaching of Computing, to ensure that all pupils are engaged and included in the subject.

It includes a number of computing-specific activities and approaches mapped to each guideline for you to try in the classroom.

The Universal Design for Learning Framework has been developed by <u>CAST</u> to improve and optimize teaching and learning for all people based on scientific insights into how humans learn.

The UDL Guidelines help educators address the diversity in learning in three main categories:

- **Engagement** (the **why** of learning): recruiting interest, sustaining effort and persistence, and self-regulation.
- **Representation** (the **what** of learning): perception, language and symbols, and comprehension.
- Action & Expression (the how of learning): physical action, expression and communication, and executive function.

For more information about UDL, please visit: <u>https://www.cast.org/what-we-do/universal-design-for-learning/</u>.

Who created the document

This document was created by the Computing & Digital Innovation Centre, part of Learn Sheffield in conjunction with Sheffield City Council. You can find more resources and discover our work to support schools with computing here: <u>https://learnsheffield.co.uk/Projects/Computing-Digital-Innovation-Centre/</u>.

Using the document

The document is free for schools to use, but please credit the <u>Computing &</u> <u>Digital Innovation Centre</u> and CAST if you repurpose any content.









This principle concerns how pupils perceive and make meaning of information, and how we, as teachers, provide options for representation.

Perception

How do you increase accessibility, flexibility, and diversity in the sharing of information?

- Support opportunities to customize the display of information (1.1)
- Support multiple ways to perceive information (1.2)
- Represent a diversity of perspectives and identities in authentic ways (1.3)

https://udlguidelines.cast.org/representation/perception/

Percention - Evamples	Key stage		e
	KS1	LKS2	UKS2
1.1 Teach pupils how to customise the display of digital content in common applications e.g. background colour, size of text.			
1.1 Teach pupils how to use common assistive technologies built into everyday digital tools to support reading and writing e.g. <u>Immersive Reader</u> , dictate/voice typing.			
1.1 When creating digital content, ask pupils to consider design choices and how this affects accessibility e.g. colour and font choice in website creation and desktop publishing.			
1.2 Use gestures to support comprehension of key concepts and vocabulary e.g. gestures to represent sequence, repetition and selection in programs.			
1.2 Provide image support for key content, e.g. using <u>image-supported glossaries</u> of key words.			
1.2 Pupils can create their own multimedia content to explain key concepts in computing e.g. films or animations to illustrate networks or variables (this also provides audience - see 7.2).			
1.3 Discuss role models in computing that reflect the diversity of identities in your class e.g. the <u>I Belong posters</u> or <u>Teaching London Computing resources</u> .			
1.3 Discuss bias in artificial intelligence and why it is important to have diverse representation in digital industries. See the <u>AI Explorers:</u> Using AI to Create New Things activity from Barefoot for content about bias in facial recognition or view the video about <u>AI: Training Data & Bias from Code.org.</u>			





This principle concerns how pupils perceive and make meaning of information, and how we, as teachers, provide options for representation.

Language & Symbols

Do you consider how language and symbols may have different meanings for pupils?

- Clarify vocabulary, symbols, and language structures (2.1)
- Support decoding of text, mathematical notation, and symbols (2.2)
- Cultivate understanding and respect across languages and dialects (2.3)
- Address biases in the use of language and symbols (2.4)
- Illustrate through multiple media (2.5)

https://udlguidelines.cast.org/representation/language-symbols/

Language & Symbols - Examples	Key stage		е
	KS1	LKS2	UKS2
2.1 Identify and teach key terms explicitly and provide opportunities for pupils to use the language in context, e.g. algorithm, networks, database.			
2.1 Provide image support for key terms, e.g. use image-supported glossaries.			
2.2 Provide pupils with plenty of practise in reading code as well as writing code. You could provide code cards (e.g. <u>for Scratch</u>) with symbol support on the reverse to aid comprehension.			
2.2 Teach pupils explicitly the commonly-used icons and symbols in the software you are using, e.g. using matching activities, reference worksheets.			
2.3 Create a multilingual book of key terms using Book Creator or PowerPoint with audio, image and text options to support learners with English as an Additional Language (EAL).			
2.3 Teach pupils to change the language in Scratch or micro:bit MakeCode to their home language if not English (in <i>Settings</i>).			
2.3 Create translation activities in Scratch to celebrate pupils' home languages, e.g. using the <i>Translate</i> and <i>text to Speech</i> extensions. See an example <u>here</u> .			
2.3 Teach pupils how to use translation tools and apps, e.g. <u>Google Translate</u> , <u>translate in Word</u> etc.			
2.5 Ask pupils to design symbols as part of a <u>Vector Graphics unit</u> to illustrate key concepts/words.			





This principle concerns how pupils perceive and make meaning of information, and how we, as teachers, provide options for representation.

Building Knowledge

How do you support learners to make connections and build their knowledge?

- Connect prior knowledge to new learning (3.1)
- Highlight and explore patterns, critical features, big ideas, and relationships (3.2)
- Cultivate multiple ways of knowing and making meaning (3.3)
- Maximize transfer and generalization (3.4)

https://udlguidelines.cast.org/representation/building-knowledge/

Many of the considerations in this guideline link to the idea of **Semantic Waves**. 'Semantic waves' describe the ideal journey for novice learners to follow when learning about abstract concepts. They move between abstract and concrete contexts, and technical and simple meanings. You can read more about them here: <u>Semantic Waves quick read</u>.

Building Knowledge - Examples	Key stage		Key stage	е
Building Knowledge - Examples	KS1	KS1 LKS2 UK		
3.1 Make explicit links between previous and current units within the curriculum, e.g. refer to what was learnt about networks in year 3 to build up knowledge about the Internet in year 4.				
3.1 Provide opportunities for retrieval practice, e.g. using multiple-choice quizzes to revisit previous content throughout the year.				
3.1 Use analogies, stories and unplugged activities to teach about new concepts and ideas, and revisit concepts in different curriculum areas, e.g. 'debug' a maths problem.				
3.2 Model how to write effective programs to pupils by live-coding. Talk about what you are doing as you go to highlight the thinking that is happening.				
3.2 Explicitly highlight the key concepts in programs, e.g. sequence, repetition, selection and variables – provide code for pupils to read and annotate, examples in different languages to compare, examples and non-examples.				
3.2 Address misconceptions and challenges that children might face e.g. the difference between the internet and the World Wide Web.				





This principle concerns how pupils perceive and make meaning of information, and how we, as teachers, provide options for representation.

Building Knowledge (contd.)

How do you support learners to make connections and build their knowledge?

- Connect prior knowledge to new learning (3.1)
- Highlight and explore patterns, critical features, big ideas, and relationships (3.2)
- Cultivate multiple ways of knowing and making meaning (3.3)
- Maximize transfer and generalization (3.4)

https://udlguidelines.cast.org/representation/building-knowledge/

Many of the considerations in this guideline link well to the idea of **Semantic Waves**. 'Semantic waves' describe the ideal journey for novice learners to follow when learning about abstract concepts. They move between abstract and concrete contexts, and technical and simple meanings. You can read more about them here: <u>Semantic Waves quick read</u>.

Building Knowledge - Examples	Key stage		е
Building Knowledge - Lkamples	KS1	LKS2	UKS2
3.3 Use unplugged activities to teach about abstract concepts, using movement, sound, objects and images.			
3.3 Provide multiple entry points and pathways through content, e.g. using <u>PRIMM</u> activities, so that pupils aren't always creating code, but they are also predicting, investigating and modifying code.			
3.4 Reduce cognitive load by using familiar contexts in unplugged activities e.g. using music to teach about sequence.			
3.4 Use semantic waves to structure activities in order to ensure maximum transfer of learning – see <u>Semantic Waves quick read</u>			
3.4 Highlight the key concepts and tools that are common across different applications and projects, e.g. save, record, debug.			





Multiple Means of Action & Expression

This principle concerns how pupils perceive and make meaning of information, and how we, as teachers, provide options for representation.

Interaction

Do pupils have the opportunity to interact with materials in different ways?

- Vary and honor the methods for response, navigation, and movement (4.1)
- Optimize access to accessible materials and assistive and accessible technologies and tools (4.2)

https://udlguidelines.cast.org/action-expression/interaction/

Interaction - Evamples	Key stage		e
	KS1	LKS2	UKS2
4.1 Use physical computing devices to teach programming as well as on-screen software e.g. Bee-Bot, Code-a-pillar, micro:bit, Crumble.			
4.1 Provide physical versions of programming commands (e.g. onlaminated <u>code</u> <u>cards</u> or 3D printed versions) for pupils to plan out their algorithms.			
4.1 Encourage pupils to act out common programming commands physically to show understanding, e.g. movement buttons on the Bee-Bot, or motion, looks and sounds blocks in Scratch.			
4.2 Teach pupils how to use common assistive technologies built into everyday digital tools to support access to content e.g. text to speech, dictate/voice typing, keyboard shortcuts.			



Computing & Digital Innovation Centre



Multiple Means of Action & Expression

This principle concerns how pupils perceive and make meaning of information, and how we, as teachers, provide options for representation.

Expression & Communication

Do pupils have the opportunity to express themselves in different ways?

- Use multiple media for communication (5.1)
- Use multiple tools for construction, composition, and creativity (5.2)
- Build fluencies with graduated support for practice and performance (5.3)
- Address biases related to modes of expression and communication (5.4)

https://udlguidelines.cast.org/action-expression/interaction/

Expression & Communication - Examples	Key stage		е
	KS1	LKS2	UKS2
5.1 Allow pupils to express themselves through multiple media, e.g. animations, films, comics, digital music and artwork. Teach how to use the software in computing lessons for cross-curricular use.			
5.1 Provide physical versions of programming commands (e.g. laminated <u>code</u> <u>cards</u> or 3D printed) for pupils to plan out their algorithms.			
5.1 Use physical objects and music in unplugged activities to teach about abstract concepts e.g. <u>Lego building algorithm</u> or <u>Musical Sequences</u> (search for SEND activities on Barefoot Computing).			
5.2 Explicitly teach learners to use text-to-speech, dictation and grammar tools to support content creation.			
5.3 Provide a number of ways to interact with code using the <u>PRIMM</u> model, with varying levels of scaffolding (p redict the outcome, r un working code, i nvestigate code, m odify an existing program, m ake your own).			
5.3 Use <u>Parson's Problems</u> to provide scaffolding for learners when programming (an activity where code is broken up and put in the wrong order, and pupils have to put it back into the correct order to make it work). There are a number of examples here in Scratch: <u>Scratch projects</u> (the <i>order</i> ones).			
5.3 Provide templates for pupils to adapt in creating media and data projects.			
5.4 Provide opportunities in creating media projects to tell stories in different ways, e.g. through animation, audio, films, comic strips.			





Multiple Means of Action & Expression

This principle concerns how pupils perceive and make meaning of information, and how we, as teachers, provide options for representation.

Strategy Development

How do you support learners to develop their executive function?

- Set meaningful goals (6.1)
- Anticipate and plan for challenges (6.2)
- Organize information and resources (6.3)
- Enhance capacity for monitoring progress (6.4)
- Challenge exclusionary practices (6.5)

https://udlguidelines.cast.org/action-expression/strategy-development/

Strategy Development - Examples	Key stage		e
	KS1	LKS2	UKS2
6.1 Ensure pupils are aware of what a successful outcome looks like by modelling good examples of digital artefacts and programs.			
6.1/6.2 Provide planning documents, including checklists, that scaffold the process for creating content or programs.			
6.2 Anticipate challenges and misconceptions and teach how to overcome these explicitly, e.g. a right turn with the Bee-Bot is a quarter-turn on the spot – practise moving like a Bee-Bot before using the technology.			
6.2 Harness challenges that pupils come across with programming as whole-class teaching opportunities, e.g. ask pupils to debug a problem encountered by a peer and talk through the solution as a class.			
6.3 When programming using a block-based language (e.g. Scratch Jr, Scratch, MakeCode on the micro:bit, Crumble), provide a help sheet of the possible code blocks required for the project and indicate where they can be found to reduce working memory demands.			
6.4 Build evaluation and reflection into planning documents (<i>Did it work? What could you improve?</i>)			





This principle concerns how pupils engage with learning and the differing motivations and interests they have.

Welcoming Interests & Identities

To what extent do you provide a level of choice and autonomy in computing activities and project work?

- Optimize choice and autonomy (7.1)
- Optimize relevance, value, and authenticity (7.2)
- Nurture joy and play (7.3)
- Address biases, threats, and distractions (7.4)

https://udlguidelines.cast.org/engagement/interests-identities/

Walcoming Interacts & Identities Examples	Key stage		е
Welconning interests & identities - Examples	KS1	LKS2	UKS2
7.1 Provide learners with a choice of context for creating media projects, e.g. poster or website design. Younger pupils may need more limited options.			
7.1 Where children are programming similar projects in Scratch Jr or Scratch, they can modify the design of backdrops and sprites to personalise their programs.			
7.1 Provide programming tasks with different levels of perceived challenge for pupils to choose between – e.g. a debugging challenge at 3 levels (e.g. cool, spicy, red hot) in Scratch.			
7.2 Identify a real-world, relevant question to answer by collecting data e.g. measuring environmental data around school using data loggers.			
7.2 Use culturally-relevant examples for projects, e.g. create a website about festivals pupils celebrate or retell traditional stories from different cultures through animation.			
7.2 Provide an audience and purpose for projects, e.g. frame digital art in a corridor, invite parents for screenings of films and animations, invite someone in to judge the best programming project.			
7.3 Tinkering is really important to foster a sense of exploration and play in computing. Provide time for pupils to explore new hardware and software and find out what it can do.			
7.4 Provide time to develop key skills such as logging on and keyboard skills to build confidence with technology.			





This principle concerns how pupils engage with learning and the differing motivations and interests they have.

Welcoming Interests & Identities (contd.)

To what extent do you provide a level of choice and autonomy in computing activities and project work?

- Optimize choice and autonomy (7.1)
- Optimize relevance, value, and authenticity (7.2)
- Nurture joy and play (7.3)
- Address biases, threats, and distractions (7.4)

https://udlguidelines.cast.org/engagement/interests-identities/

Welcoming Interests & Identities - Evamples	Key stage		Key stage		е
Welcoming interests & identities - Examples	KS1	LKS2	UKS2		
7.4 Develop strategies to reduce the distraction of technology in the classroom, for example modelling activities away from computers, turning off screens or turning tablets upside down when listening.					
7.4 Use apps like <u>Plickers</u> to reduce the threat when answering questions (see our video <u>here</u> on how to use the tool).					
7.4 Provide templates (e.g. for a poster or pictogram) or working programs for pupils to modify in order to reduce the threat of failure. See our collection of <u>Scratch projects</u> that can be used for this.					
7.4 Teach debugging and problem-solving approaches to pupils and celebrate a culture of making mistakes – model this in your own practice when things go wrong!					





This principle concerns how pupils engage with learning and the differing motivations and interests they have.

Sustaining Effort & Persistence

How do you support learners to sustain effort in tasks?

- Clarify the meaning and purpose of goals (8.1)
- Optimize challenge and support (8.2)
- Foster collaboration, interdependence, and collective learning (8.3)
- Foster belonging and community (8.4)
- Offer action-oriented feedback (8.5)

https://udlguidelines.cast.org/engagement/effort-persistence/

Sustaining Effort & Persistence - Evamples	K	Key stage	
	KS1	LKS2	UKS2
8.1 Provide examples of good content and code and identify with pupils what makes it good.			
8.1 Use detailed planning documents to guide pupils to fulfil given goals.			
8.2 Provide different ways to interact with code using the <u>PRIMM</u> model and different levels of challenge (e.g. cool, spicy, red hot programming challenges).			
8.3 Use the <u>Pair Programming</u> approach to support learners to work together on one device – this is a driver/navigator model where one person controls the computer, the other provides instructions, and then they swap over.			
8.3 Provide very specific instructions to pairs and groups for collaborative work, e.g. when using Bee-Bots one person can write the algorithm, one person inputs the code on the Bee-Bot and one person evaluates the success and debugs.			
8.3 Use FAQs to support peers – the first person who asks for help gets instruction from the teacher. They are then assigned as expert and their name goes on the board next to the issue that was solved (the FAQ). The next person who needs help asks the expert – and then becomes the next expert and so on.			
8.4 Celebrate achievements and showcase finished projects, e.g. a digital art gallery, blog of films etc.			
8.5 Build self-evaluation into planning documents for projects.			





This principle concerns how pupils engage with learning and the differing motivations and interests they have.

Emotional Capacity

Do you support pupils to develop emotional capacity and coping strategies?

- Recognize expectations, beliefs, and motivations (9.1)
- Develop awareness of self and others (9.2)
- Promote individual and collective reflection (9.3)
- Cultivate empathy and restorative practices (9.4)

https://udlguidelines.cast.org/engagement/emotional-capacity/

Emotional Canacity - Examples	Key stage		е
	KS1	LKS2	UKS2
9.1 Provide checklists for pupils to follow to complete more complex projects, in order to help keep them on task.			
9.1 Model resilience when things go wrong and think aloud when applying problem-solving approaches e.g. turn it off and on again!			
9.2 Teach problem-solving and debugging strategies to pupils, and emphasise the role and importance of making mistakes in computing.			
9.2 Discuss the different roles and careers in computing, and the range of skills required e.g. composing the music or drawing the backgrounds for a game as important as the coding.			
9.4 Co-create rules for collaborative work and use the FAQs model for peer- support (see 8.3).			

