Intent, Implementation and Impact in Computing

'Working together, learning together, achieving together'

At Mattishall Primary School we value the whole child and balance their academic needs with their social, emotional and personal development. We nurture children to have active, inquisitive and creative minds. We help them by:

- ***providing a high-quality curriculum with a clear pedagogical approach***
- ***developing empathy, confidence and resilience***
- ***recognising what equality, diversity and tolerance means***
- ***equipping them with 'life skills' and behaviours for learning***
- ***encouraging individuality***
- ***having high expectations and celebrating success and achievement***
- ***raising aspirations for the present and future***
- ***providing a stimulating environment***
- ***promoting a positive partnership with our parents/carers***
- ***developing independent global citizens of the future***

Intent

At Mattishall our Vision for Computing is to equip all of our pupils to become digitally literate and to use computational thinking and creativity to create programs and systems to further understand advances in technology and our ever-changing world. We aim to equip our pupils with the knowledge and skills to stay safe online.

Implementation

The Mattishall Primary Computing curriculum has been mapped to reflect a spiral curriculum. The units for key stages 1 and 2 are based on a spiral curriculum. This means that each of the themes is revisited regularly (at least once in each year group), and pupils revisit each theme through a new unit that consolidates and builds on prior learning within that theme. The children are taught key computational skills such as how to become efficient programmers, to understand computer systems, networks and the internet. To understand how to handle data, how to create different types of media and how to stay safe online.

We teach computing using 12 pedagogy principles and use teach computing.org.uk resources.

The 12 pedagogy principles we use are:

1. Lead with concepts

Support pupils in the acquisition of knowledge, through the use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps, and displays, along with regular recall and revision, can support this approach.

2. Work together

Encourage collaboration, specifically using pair programming and peer instruction,
and also, structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding.

3. Get hands-on

Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides pupils with a creative, engaging context to explore
and apply computing concepts.

4. Make concrete

Bring abstract concepts to life with real-world, contextual examples and a focus on interdependencies with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in pupils' lives.

5. Create projects

Use project-based learning activities to provide pupils with the opportunity to apply and consolidate their knowledge and understanding. Design is an important, often overlooked aspect of computing. Pupils can consider how to develop an artefact for a particular user or function, and evaluate it against a set of criteria.

6. Add variety

Provide activities with different levels of direction, scaffolding, and support that promote active learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all pupils engaged and encourage greater independence.

7. Structure lessons

Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and Use-Modify-Create. These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.

8. Read and explore code first

When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and text-based programming, encourage pupils to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments pupils' ability to write code.

9. Challenge misconceptions

Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.

10. Unplug, unpack, repack

Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach (semantic waves) can help pupils develop a secure understanding of complex concepts.

11. Model everything

Model processes or practices — everything from debugging code to binary number conversions — using techniques such as worked example s and live coding. Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away.

12. Foster program comprehension

Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's Problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.

Impact

We have different ways of measuring the success of learning across the curriculum. This allows children to celebrate their successes as well as knowing what they need to do to progress. These may include:

Statutory assessment (Maths, English)
Adult observation including staff, parent/carers and governors
Self-Assessment (Traffic light)
Attainment Tracker
Recorded Tasks (Children's work)
Verbal Feedback
End of unit quizzes
Vocab Victories
Video /photo evidence
Performance
Talking Partners
Peer feedback
Pupil Progress Tracking meetings
Pupil Passport (SEND)