LIVERPOOL JOHN MOORES UNIVERSITY

Guidance for Observation and Target Setting in Secondary Computer Science

National Curriculum Purpose of study for computing aims ensure that all children: can understand and apply the principles and concepts of computer science, including abstraction, logic, algorithms and data representation. Computing has deep links with mathematics, science and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work and how to put this knowledge to use through programming.

6 Key Questions to ask when observing Computer Science lessons:

	Question	Additional Information
1.	Are pupils encouraged to investigate and experiment, being guided to reflect on what they discover?	*For example, are wider concepts like data validation and functions brought into programs to extend more able pupils?*
2	Is there opportunity for collaborative work?	*Are on-line collaborative coding packages being utilised? Or could pupils test and comment on each other's code?*
3	Is computational thinking (a way of looking at problems and systems so that computers can be used to help solve or understand them) promoted in these lessons?	*For example, could pupils be encouraged to consider the different ways that they interact with computer programs in their everyday life?*
4	Have any risks regarding online safety been assessed? Are pupils taught how to keep themselves safe and act responsibly online?	*Is current legislation and general school activity tied into the teaching of the ethical, legal, cultural and environmental impact unit?*
5	Does the teacher have a high level of competence and expertise, both in terms of their specialist knowledge and technical skills and in their understanding of active learning in computing?	*For example, are opportunities for self-development CPD explored? (for example Code Academy, SoloLearn learning packages)*
6	Do pupils develop links across the two main Computing Curriculum Topic Areas?	*For example, could coding projects incorporate aspects of the Unit 1 Computer Systems theory? (E.g a quiz based on topics covered in Networks, or Systems Architecture?)*

LIVERPOOL JOHN MOORES UNIVERSITY

Potential Computer Science Specific Targets on Lesson Analysis Forms.

Lesson design and delivery, including sequencing and choice of teaching methods (CCF curriculum & pedagogy) Next Steps:
Consider the ways that the Computer Science curriculum builds; from systems architecture, data representation, operating system, networks through to social and environmental issues. Attempt to link current curriculum areas with previous and future ones to allow pupils to visualise a linked chain of process.
Use analogy to break down some of the more abstract or unseen concepts of computing systems architecture.
Make use of worked examples of code to allow weaker (or absent) pupils to access curriculum content fully and at a pace more suited to them. Also consider ways this code can be structured (for example, partially completed code with strategically placed a gaps for the pupils to fill)
Consider pupils prior experience in KS3 when designing KS4 lessons and how links between the key stages can be developed. For instance, drawing parallels between the code in Scratch and Python.
Question pupils thoroughly to develop a class-wide understanding of the curriculum. Encourage *them* as much as possible to provide the explanations. Move the conversation around the class. For example, one pupil provides an advantage and another pupil a disadvantage of a given scenario.

Pupil progress in this lesson and use of assessment (including questioning) (CCF assessment) Next Steps:
Consider ways that feedback can encourage pupils to solve programming problems themselves as opposed to outright solving the code for them.
Plan opportunities for pupils to use analogy in their oral descriptions of Computer Code. (For example, encourage them to describe the events within a program like a story)
Encourage pupils to identify and utilise the guidance and messages provided by their chosen programming IDE. Thus enabling them to become more independent learners who can identify and solve problems unaided.
Plan opportunities for pupils to investigate how the network in their school setting (or home) operates. (For example, a homework investigating their IP Address, Router Security, Bandwidth etc)
Develop opportunities for pupils to recall prior knowledge. Revisiting and developing concepts previously learned. Consider starting each lesson with quick recall questions from previous topics.

Comments about student teacher's developing Subject Knowledge and Pedagogy (CCF curriculum & pedagogy) Next Steps for further development
Develop an understanding of the key programming concepts outlined in National Curriculum and specifically the OCR Unit 2 Computational Thinking unit.
Develop skills in identifying common mistakes and misconceptions in pupil's programming, and also strategies that the pupils could employ to identify and solve these problems

themselves.
Develop an awareness of the range of resources available to Computer Science teachers. Specially the Teaching Computing' website.
Enrol with the CAS (Computing At Schools) group and sign up for their forum bulletins. Similarly, sign up to the Facebook group for OCR Computer Science teachers. This will help to develop subject knowledge, pedagogy and also the ever-changing trends and options related to the profession.
Develop an understanding of potential cross-curricular links between the Computer Science Curriculum and other lessons that pupils undertake. Especially links with Maths between data representation, logical programming concepts, sorting and sequencing and binary logic gates.