The LJMU Mentor Guide to the curriculum in Phase 3
Secondary COMPUTING

The ITE Curriculum in Phase 3 further extends student teachers' critical engagement with research and practice, within the relevant subject(s) and age phase(s). Most of Phase 3 is school based, as the student teacher completes their final placement, extending their knowledge and skills to achieve a recommendation for Qualified Teacher Status (QTS).

The Phase 3 curriculum (centre based) includes a focus on:

- Subject pedagogies and strategies across the Secondary age range.
- Curriculum design.
- Transition between key stages.
- Statutory assessment.
- Deepening Subject knowledge and curriculum knowledge.
- Adaptive practice.
- Preparing for the ECT year.

School based training:

During Phase 3, the ITE curriculum will be extended and contextualised within school placements.

Training will enable student teachers to successfully demonstrate a range of teaching approaches, appropriate to subject, age, and individual needs, and contribute to the development of curriculum thinking, drawing on evidence from research.

Whilst we expect the ITE curriculum to be increasingly personalised during Phase 3, in order to meet individual areas for development and school priorities, it is likely that student teachers' understanding and experience of working with other adults (including parents), and of assessment and adaptive teaching will need deepening during this final period, supported by expert colleagues.

During Phase 3 student teachers have more independent responsibility for planning and teaching lessons which are well matched to the needs of groups and individuals, and for planning units of work as part of an ambitious curriculum, with a positive impact on pupil progress.

Expected outcomes

Mid-point and end of Phase reviews are completed by ITT Mentors in schools. In Phase 3, these reviews assess progress against the Teachers' Standards. By the end of Phase 3 we expect student teachers to:

- Compile evidence in placement experience folder to demonstrate achievement of ALL of the Teachers' Standards (including Part 2)
- Respond constructively to challenge, feedback and critique, and demonstrate a commitment to continuously improve their understanding and practice, reflected in their Career Entry and Development Profile.

In Computer Science sessions:

Please see the ITE Curriculum for Computer Science for the full programme throughout the year.

The main focus of this phase is to ensure that the student teacher is confident in teaching all aspects of the curriculum. Building on initial theories of learning and classroom management the focus orientates towards practical methods of designing, delivering and assessing lessons. Previously taught pedagogical concepts taught in phase 2 are recalled and embedded into the classroom setting that the trainees are becoming increasingly familiar with. Common pupil misconceptions with aspects of the curriculum are also addressed, to ensure trainees have the confidence to resolve technical problems they may encounter.

Some sessions with look beyond the classroom at wider aspects of the school experience and current research trends and possible future developments within the ever changing Computer Science curriculum.

The Phase 3a ITE Computer Science Curriculum:

In Phase 3, the centre-based curriculum focuses on subject knowledge, pedagogy, adaptive practice, assessment and progress in the subject. We ask you to support students in exploring these further in schools.

The focus of weekly discussions is in black. Professional Development Activities for STUDENTS are in blue.

Date (LJMU)	Taught LJMU session	School-based focus	Mentor curriculum in weekly meeting and Professional Development Activities.
Friday 23 FEB	Cross-curricular Computer Science teaching opportunities.	Consider ways that Computer Science can support literacy and PHSE lessons. For example, are there any aspects of the Ethical, Legal, Cultural and Environmental aspects of the Computer Science curriculum taught elsewhere in the school?	Student teacher: Look at other subject's curriculum plans for specific references to topic that overlap with the Computer Science curriculum and note the type of content and activities that are used. Can your Computing lessons be built and expand on prior knowledge? Mentor: Provide access to the *other* subjects curriculum plans, so that the trainee can see where complimentary topics are taught.
Friday 1 MAR	Subject Knowledge Development Lesson covering the KS4 Computer Science Curriculum	Following on from this session, the focus should be to develop practical skills and pedagogical practice in the delivery of the Computer Science Data Representation unit.	Student teacher: Look at existing teaching material within the school and consider how this would be best utilised or adapted to deliver the data representation unit. Mentor: discuss teaching strategies with the student teacher on the delivery of the Data Representation unit, including martials used and common misconceptions within the area.
Friday 8 MAR	Subject Knowledge Development Lesson covering the KS4 Computer Science Curriculum	Following on from this session, the focus should be to develop practical skills and pedagogical practice in the delivery of the Computer Science Networks unit.	Student teacher: Look at possible ways that you could assist in showing pupils physical parts of the school network. Mentor: Attempt to introduce the student teacher to the school's network manager with a view to them assisting in a series of small tours / talk about the school network (e.g visit to the server / switch room) Or consider how the student teacher could photograph this part of the school network and a result help pupils to further understand the theory.
Friday 15 MAR	Subject planning: Designing a scheme of work.	The session will look at strategies for how to break down and interpret exam board specification.	Student teacher: Look at your schools KS4 Computer Science scheme of work and map it to the OCR Computer Science specification. Look at how (and reflect on why) the exam board specification is

			spread across both year 10 and year 11. Menor: Provide the student teacher with a copy of the KS4 Computer Science fully year scheme of work and provide a rational as to why the topics are taught in the order the school has chosen.
Friday 22 MAR	GCSE Exam Paper Breakdown Session 1 / 2 GCSE Exam Paper Breakdown Session 2 / 2	Identify the links between GCSE computing topics with KS3 NC and A'level curriculum expectations.	Student teacher: Complete a GCSE exam paper and use the assessment criteria / scheme to mark accurately. Read the most recent chief examiners report to understand where pupils are most successful and where they need most support in relation to the GCSE specification. Mentor: discuss the GCSE paper and outcome the student teacher completed, as well as their understanding of the chief examiners report. Support them to create an action plan for their own subject knowledge development at GCSE and advise on relevant resources to support.
Friday 29 MAR	Supporting SEND pupils	A look at the ways to make computing lessons can support for learners with special educational needs and disabilities.	Student teacher: Meet with the school SEND staff and discuss ways in which the school supports pupils across *all* subjects. Mentor: Provide the student teacher with examples of how teaching is differentiated to support all learners. For example, are class sets decided on ability, does lesson content and expectations differ across the class sets?
Friday 26 APR	Subject Knowledge Development Lesson covering the KS4 Computer Science Curriculum	Providing opportunity to draw upon existing pedagogy in how to teach the Java programming language.	Student teacher: Find out if the school has previously, does presently or intends in the future to teach programming languages *other* than Python. Mentor: Discuss ways in which your staff re-train and learn new skills to meet the ever changing demands of teaching the computer science curriculum.

Lesson observation focus questions: these questions can be used as foci or prompt for lesson observations and mentor/student teacher discourse. Not all elements are required in every lesson and may be phase dependent, i.e., expectations that more elements would be present in phase three of teaching as confidence and highly effective practice is embedded.

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?)*

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.