Assessment at KS3

Rationale:

Effective student assessment is of paramount importance when it comes to student progress.

Students need to know 'where they are' in relation to their understanding and 'how they can improve'.

Teachers need to be able to consistently assess students understanding so that they can provide effective, tailored feedback and pitch lessons appropriately to so that all students within a lesson make sustained and rapid progress.

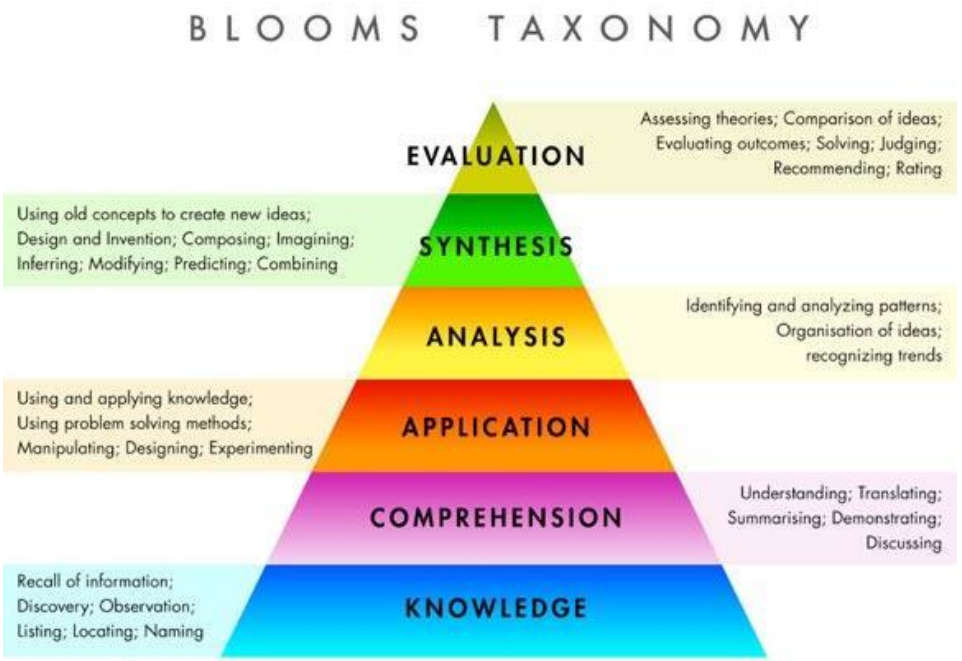
Skills and Understanding (Blooms Taxonomy) VS Content and Knowledge (Computing at School's 'Knowledge Pathway Grid'):

Since the current Computing Curriculum has no set level descriptors we have decided to create our own assessment which is tailored to the philosophy of ComputerScienceUK.

When deciding on how we would assess students within the various areas of the subject we first studied the various methods discussed on the 'Computing at School's Website'. A widely accepted method is basing assessment on the 'Progression Pathways Assessment Framework KS1 (Y1) to KS3 (Y9)' (Mark Dorling 2014). The issue we had with this is that despite the grid giving a clear frame work for progression in knowledge and content (see grid on next page), it doesn't necessarily assess their level of understanding for these topics. At ComputerScienceUK, our philosophy when it comes to assessment is that assessing knowledge needs to be carefully balanced with understanding. A student may have excellent recall for examinations but not necessarily have a deep understanding of the content being assessed.

We therefore decided to investigate methods of assessing 'understanding' which led us to examine Bloom's Taxonomy and decided to incorporate this theory of 'higher order questioning' when assessing students at the end of each unit.

It was therefore decided that the 'progression pathways grid' would be used to decide the content to be delivered in each unit (e.g. year 8 hardware unit would roughly cover content from the hardware column in rows levelled 4-7) and the subsequent topic areas to be assessed (so that students' knowledge of these areas could be assessed) and that Blooms Taxonomy would be used to set the various styles of questions for each of these topic areas (so that student's understanding could be assessed).



Computer Science UK
...for teachers and students alike

Computing Progression Pathways *(Mark Dorling 2014)*

	Computer Science Strand					Digital Creativity, Citizenship and Literacy Strand
Level	**Algorithms**	**Programming & Development**	**Data & Data Representation**	**Hardware & Processing**	**Communications & Networks**	**Digital Creativity, Digital Citizenship & Digital Literacy**
1	Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. Understands that computers need precise instructions. Demonstrates care and precision to avoid errors.	Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text e.g. programmable robots etc. Executes, checks and changes programs. Understands that programs execute by following precise instructions.	Recognises that digital content can be represented in many forms. Distinguishes between some of these forms and can explain the different ways that they communicate information.	Understands that computers have no intelligence and that computers can do nothing unless a program is executed. Recognises that all software executed on digital devices is programmed.	Obtains content from the world wide web using a web browser. Understands the importance of communicating safely and respectfully online, and the need for keeping personal information private. Knows what to do when concerned about content or being contacted.	Uses software under the control of the tea to create, store and edit digital content us appropriate file and folder names. Understand that people interact with computers. Share their use of technology in school. Knows common uses of information technology beyond the classroom. Talks about their w and makes changes to improve it.
2	Understands that algorithms are implemented on digital devices as programs. Designs simple algorithms using loops, and selection i.e. if statements. Uses logical reasoning to predict outcomes. Detects and corrects errors i.e. debugging, in algorithms.	Uses arithmetic operators, if statements, and loops, within programs. Uses logical reasoning to predict the behaviour of programs. Detects and corrects simple semantic errors i.e. debugging, in programs.	Recognises different types of data: text, number. Appreciates that programs can work with different types of data. Recognises that data can be structured in tables to make it useful.	Recognises that a range of digital devices can be considered a computer. Recognises and can use a range of input and output devices. Understands how programs specify the function of a general purpose computer.	Navigates the web and can carry out simple web searches to collect digital content. Demonstrates use of computers safely and responsibly, knowing a range of ways to report unacceptable content and contact when online.	Uses technology with increasing independence to purposefully organise digital content. Shows an awareness for the quality of dig content collected. Uses a variety of software to manipulate and present digital content: and information. Shares their experiences of technology in school and beyond the classroom. Talks about their work and ma improvements to solutions based on feed received.
3	Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. Uses diagrams to express solutions. Uses logical reasoning to predict outputs, showing an awareness of inputs.	Creates programs that implement algorithms to achieve given goals. Declares and assigns variables. Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.	Understands the difference between data and information. Knows why sorting data in a flat file can improve searching for information. Uses filters or can perform single criteria searches for information.	Knows that computers collect data from various input devices, including sensors and application software. Understands the difference between hardware and application software, and their roles within a computer system.	Understands the difference between the internet and internet service e.g. world wide web. Shows an awareness of, and can use a range of internet services e.g. VOIP. Recognises what is acceptable and unacceptable behaviour when using technologies and online services.	Collects, organises and presents data and information in digital content. Creates dig content to achieve a given goal through combining software packages and interne services to communicate with a wider audience e.g. blogging. Makes appropriate improvements to solutions based on feed received, and can comment on the success the solution.
4	Shows an awareness of tasks best completed by humans or computers. Designs solutions by decomposing a problem and creates a sub-solution for each of these parts (decomposition). Recognises that different solutions exist for the same problem.	Understands the difference between, and appropriately uses if and if, then and else statements. Uses a variable and relational operators within a loop to govern termination. Designs, writes and debugs modular programs using procedures. Knows that a procedure can be used to hide the detail with sub-solution (procedural abstraction).	Performs more complex searches for information e.g. using Boolean and relational operators. Analyses and evaluates data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions.	Understands why and when computers are used. Understands the main functions of the operating system. Knows the difference between physical, wireless and mobile networks.	Understands how to effectively use search engines, and knows how search results are selected, including that search engines use 'web crawler programs'. Selects, combines and uses internet services. Demonstrates responsible use of technologies and online services, and knows a range of ways to report concerns.	Makes judgements about digital content when evaluating and repurposing it for a given audience. Recognises the audience when designing and creating digital content Understands the potential of information technology for collaboration when computers are networked. Uses criteria to evaluate the quality of solutions, can identify improvement making some refinements to the solution, future solutions.
5	Understands that iteration is the repetition of a process such as a loop. Recognises that different algorithms exist for the same problem. Represents solutions using a structured notation. Can identify similarities and differences in situations and can use these to solve problems (pattern recognition).	Understands that programming bridges the gap between algorithmic solutions and computers. Has practical experience of a high-level textual language, including using standard libraries when programming. Uses a range of operators and expressions e.g. Boolean, and applies them in the context of program control. Selects the appropriate data types.	Knows that digital computers use binary to represent all data. Understands how bit patterns represent numbers and images. Knows that computers transfer data in binary. Understands the relationship between binary and file size (uncompressed). Defines data types: real numbers and Boolean. Queries data on one table using a typical query language.	Recognises and understands the function of the main internal parts of basic computer architecture. Understands the concepts behind the fetch-execute cycle. Knows that there is a range of operating systems and application software for the same hardware.	Understands how search engines rank search results. Understands how to construct static web pages using HTML and CSS. Understands data transmission between digital computers over networks, including the internet i.e. IP addresses and packet switching.	Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals. Recognise ethical issues surrounding the application of information technology beyond school Designs criteria to critically evaluate the quality of solutions, uses the criteria to ide improvements and can make appropriate refinements to the solution.
6	Understands a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. Recognises that some problems share the same characteristics and use the same algorithm to solve both (generalisation). Understands the notion of performance for algorithms and appreciates that some algorithms have different performance characteristics for the same task.	Uses nested selection statements. Appreciates the need for, and writes, custom functions including use of parameters. Knows the difference between, and uses appropriately, procedures and functions. Understands and uses negation with operators. Uses and manipulates one dimensional data structures. Detects and corrects syntactical errors.	Understands how numbers, images, sounds and character sets use the same bit patterns. Performs simple operations using bit patterns e.g. binary addition. Understands the relationship between resolution and colour depth, including the effect on file size. Distinguishes between data used in a simple program (a variable) and the storage structure for that data.	Understands the von Neumann architecture in relation to the fetch-execute cycle, including how data is stored in memory. Understands the basic function and operation of location addressable memory.	Knows 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. Uses technologies and online services securely, and knows how to identify and report inappropriate conduct.	Justifies the choice of and independently combines and uses multiple digital device internet services and application software to achieve given goals. Evaluates the trustworthiness of digital content and con the usability of visual design features when designing and creating digital artefacts for known audience. Identifies and explains how the use of technology can impact on society. Designs criteria for users to evaluate the quality of solutions, uses the feedback from users to identify improvements and can make appropriate refinements to the solution.
7	Recognises that the design of an algorithm is distinct from its expression in a programming language (which will depend on the programming constructs available). Evaluates the effectiveness of algorithms and models for similar problems. Recognises where information can be filtered out in generalizing problem solutions (abstraction). Uses logical reasoning to explain how an algorithm works. Represents algorithms using structured language.	Appreciates the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function. Understands and applies parameter passing. Understands the difference between, and uses, both pre-tested e.g. 'while', and post-tested e.g. 'until' loops. Applies a modular approach to error detection and correction.	Knows the relationship between data representation and data quality. Understands the relationship between binary and electrical circuits, including Boolean logic. Understands how and why values are data typed in many different languages when manipulated within programs.	Knows that processors have instruction sets and that these relate to low-level instructions carried out by a computer.	Knows the purpose of the hardware and protocols associated with networking computer systems. Understands the client-server model including how dynamic web pages use server-side scripting and that webservers process and store data entered by users. Recognises that persistence of data on the internet requires careful protection of online identity and privacy.	Undertakes creative projects that collect, analyse, and evaluate data to meet the ne of a known user group. Effectively design creates digital artefacts for a wider or rem audience. Considers the properties of med when importing them into digital artefacts Documents user feedback, the improvements identified and the refinements made to the solution. Explains and justifies how the use of technology impacts on society, from the perspective of social, economic, political, legal, ethical and moral issues.
8	Designs a solution to a problem that depends on solutions to smaller instances of the same problem (recursion). Understands that some problems cannot be solved computationally.	Designs and writes nested modular programs that enforce reusability utilising sub-routines wherever possible. Understands the difference between 'While' loop and 'For' loop, which uses a loop counter. Understands and uses two dimensional data structures.	Performs operations using bit patterns e.g. conversion between binary and hexadecimal, binary subtraction etc. Understands and can explain the need for data compression, and performs simple compression methods. Knows what a relational database is, and understands the benefits of storing data in multiple tables.	Has practical experience of a small (hypothetical) low level programming language. Understands and can explain Moore's Law. Understands and can explain multitasking by computers	Understands the hardware associated with networking computer systems, including WANs and LANs, understands their purpose and how they work, including MAC addresses.	Understands the ethical issues surrounding application of information technology, an existence of legal frameworks governing its use e.g. Data Protection Act, Computer Misuse Copyright etc.

Conclusion

Knowledge and Understanding are not the same.

Students may be able to remember and recall knowledge but will not necessarily understand it.

The grid above clearly shows progression in terms of content/topics and their increasing difficulty but the issue is that this is very much 'knowledge based' and so using this grid alone to assess students will not fully assess an individual's ***understanding*** of the various aspects of the curriculum.

It is ComputerScienceUK's aim to try to 'marry together' the 'Progression Pathways Grid' with 'Blooms Taxonomy' to assess:

- students' knowledge of the content delivered for each unit (by setting questions which cover the topics from the unit which in turn have been pitched over a range of levels)
- students' relative understanding of the content delivered (by setting a range of question styles from the more simple: 'list', 'define' etc. to the harder: 'explain' or 'discuss').

Assessment of Programming Units	**Assessment of Computer Science Theory Units**	**Assessment of Digital Creativity** ***(inc. DL and DCitz)***
End of Unit Examination Content taught in these units will be assessed with a series of problem solving tasks. These tasks will require the application of the content covered in the unit but will get progressively harder as the required autonomy needed to solve each problem increases. Using this method, it is not just content / knowledge which will be assessed, in addition their strength of understanding will be measured. Year 7: 'Content levelled 3-6' will be taught and assessed Year 8: 'Content levelled 4-7' will be taught and assessed Year 9: 'Content levelled 5-8' will be taught and assessed	End of Unit Examination Content taught in these units will be assessed with a series of exam questions. These questions will require the application of the content covered in the unit but will get progressively harder as the question style progresses from the more simple: 'list', 'define' etc. to the harder: 'explain' or 'discuss'. Using this method, it is not just content / knowledge which will be assessed, in addition their strength of understanding will be measured. Year 7: 'Content levelled 3-6' will be taught and assessed Year 8: 'Content levelled 4-7' will be taught and assessed Year 9: 'Content levelled 5-8' will be taught and assessed	Extended Project Work Projects produced will be assessed based on the Knowledge Pathways Grid. During these units the various stages of the systems development life cycle will be focused upon and during these lessons students will use success criteria and assessment criteria to help show them how to assess how good their work is and how it can be improved. These grids are later used to assess their final work so that assessment is consistent across the department.

Computer Science Theory: *Assessment Grid*

Analysis: Links

Applying: Explanations, Discussions, Links

Understanding: (In-Depth) Descriptions and Explanations

Knowledge: Stating, Listing, Identification, (Superficial) Description

	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8
Computer Science Theory	Can correctly answer **some** "State", "Label", "List", "Identify" theory questions.	Can correctly answer **most** "State", "Label", "List", "Identify" theory questions and **some** "Describe" questions.	Can correctly answer "State", "Label", "List", "Identify" theory questions and **most** "Describe" and **some** "Explain" questions.	Can correctly answer "State", "Label", "List", "Identify" and "Describe" theory questions and **most** "Explain" questions.	Can correctly answer "State", "Label", "List", "Identify", "Describe" theory questions, **most** "Explain" questions and **some** "Discuss" questions.	Can correctly answer "State", "Label", "List", "Identify", "Describe" theory questions, **most** "Explain" and "Discuss" questions. Is starting to be able use 'current understanding' to access higher level content (links, themes, patterns etc.)

Example of higher order questioning (hardware example):

L3: Label some of the components a computer.
L4: Label the components of a computer and describe at least one component's purpose.
L5: Describe the purpose of the major components of a computer.
L6: Describe the purpose of the components of a computer and explain how at least one of the components works.
L7: Explain how the major components of a computer work.
L8: Discuss how technological developments of a major component may impact the performance of future computers.

So ***in summary***, for a given year's exam, students will be asked questions on topics pitched across a range of levels (based on knowledge pathways grid) to assess their level of knowledge. And for each topic, several questions will be asked which range from the more simple style (list, define) to the more challenging style (explain, discuss) in order to assess their level of understanding (based on Bloom's Taxonomy).

Programming: *Assessment Grid*

Evaluate: Highly Independent

Synthesis: Use Knowledge to Design / Invent

Analysis: Solving Problems (independently) and Programming Solutions

Applying: Solving Problems (with support) and Programming Solutions (independently)

Understanding: Programming (prescribed) solutions (using own knowledge)

Knowledge: Replicate (following tutorials), Recall

	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8
Programming	Sequence instructions (with guidance). Needs problems broken down for them.	Can sequence instructions with some independence. Shows some understanding of the concepts of inputs, outputs, variables and but needs some help implementing them Shows some understanding of the control structures of selection and iteration but requires some support when applying them in order to solve basic programming tasks. Can break down simple problems with support in order to design suitable solutions.	Can sequence instructions with independence. Shows understanding of the concepts of inputs, outputs, variables and can implement them with little support. Shows understanding of the control structures of selection and iteration and can implement with little support but requires more support when applying them in order to solve harder problems. Can break down simple problems independently in order to design suitable solutions.	Can sequence instructions with independence. Shows understanding of the concepts of inputs, outputs, variables and simple functions and can implement them independently. Shows understanding of the control structures of selection and iteration and can implement them independently but requires more support when applying them in order to solve harder problems. Can break down harder problems with support in order to design suitable solutions.	Can sequence instructions with independence. Shows understanding of the concepts of inputs, outputs, variables and simple functions and can implement them independently. Shows understanding of the control structures of selection and iteration and can implement them independently but requires more support when applying them in order to solve harder problems. Can break down harder problems with little support in order to design suitable solutions.	Can sequence instructions with independence. Shows understanding of the concepts of inputs, outputs, variables and some more advanced functions and can implement them independently. Shows understanding of the control structures of selection and iteration and can implement them independently even when applying them in order to solve harder problems. Can break down harder problems with independence in order to design suitable solutions.

Computer Science UK
...for teachers and students alike

Example of progression of problems to be solved:

Level 3/4 Questions

1. Which of the following pieces of code will simply display a message once, on the screen, **without** waiting for the user to type in a response?

```
name = input("What is your name?)
```

2. Which of the following pieces of code will display a message on the screen and store the response in a variable?

```
for x in range (10):
    print("apples")
```

```
print("Hello World")
```

3. Which of the following code will program decisions?

```
if mood == "good":
    print("You're happy")
else:
    print("You're not happy")
```

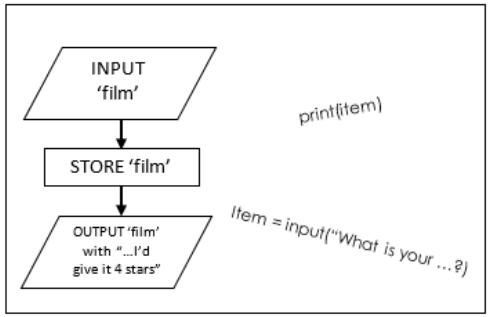
4. Which of the following pieces of code will display a word 10 times on the screen?

```
While x == 0:
    print("Hello World")
```

Level 5 Questions

5. By using, adapting and ordering the code shown opposite and using the flowchart as a guide, create a program which:
   - Gets the program to ask the user to type in their favourite film
   - Displays the film that the user just typed in along with the text "...I'd give it 4 stars!!"

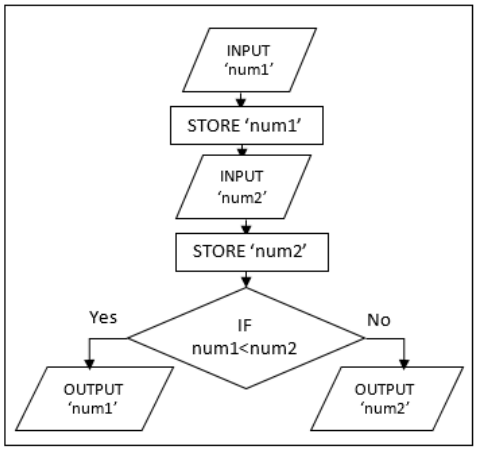
   Paste a screen shot of your code in the box below – make sure it is cropped and of a good size so that I can read it!



Level 6 Questions

7. Using the flowchart as a guide, create a program which:
   - Gets the program to ask the user to type in 2 numbers
   - The program must then decide which is the smallest of the two numbers and display the biggest number on the screen

Paste a screen shot of your code in the box below – make sure it is cropped and of a good size so that I can read it!



Level 7 Questions

Create a 'totaliser' program which:

- Gets the program to repeatedly ask the user to enter numbers
- Each time a number is entered it is added to the total
- The total is then displayed.
- This continues until the user closes the program.

Paste a screen shot of your code in the box below – make sure it is cropped and of a good size so that I can read it!