Example Graduated Computing Statements for Assessment for Learning Without Levels

KS2 Examples Using **Identify Develop Explain** or ***Imitate Adapt Create***

Pupils should be taught to design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts			
	Identify	Develop	Explain
Design programs that accomplish specific goals	I can identify that programs on digital devices start as algorithms I can identify some different ways programs can be planned as algorithms • Flowcharts • Normal language • Pseudo code	I can design an algorithm that accomplishes a specific goal that could be converted into code I can use a flowchart or plain English to design an algorithm for a specific goal	I can explain the steps or parts of my algorithm and how they could be turned into code I can explain why a specific algorithm method was useful in designing my program I can explain the purpose of my algorithm
	Imitate	Adapt	Create
Design programs that accomplish specific goals	I can examine an algorithm and answer questions about its steps, parts and purpose	I can complete an unfinished algorithm I can adapt an algorithm to solve a similar problem	I can design an algorithm that accomplishes a specific goal that could be converted into code I can use a flowchart or normal language to design an algorithm for a specific goal
	Identify	Develop	Explain
Write programs that accomplish specific goals	I identify that writing a program is turning an algorithm into code I can identify the programming language I am using	I can convert an algorithm into a program that accomplishes a specific goal I can identify other programming languages that could be used	I can document my program listing • Parts of the algorithm that were difficult to convert • Alternate solutions and why the final one was chosen • Possible program enhancements

	Imitate	Adapt	Create
Write programs that accomplish specific goals	I can examine a program and answer questions about its steps, parts and purpose I can follow instruction to build a program	I can complete an unfinished program I can adapt a program to solve a similar problem	I can write a program by converting an algorithm into code
	Identify	Develop	Explain
Debug programs	I can identify that • All programs will have bugs in them • Bugs are normal in programming • It may take time to find and fix my bug • I am responsible for debugging my own code	I can systematically debug my programs by • Explaining what my code should do to a neighbour • Running just a part of the code • Reading my code out loud • Going back to my algorithm and comparing it to the code	I can explain how I debugged my code, what strategies I used and what I learnt from the process
	Imitate	Adapt	Create
Debug programs	I can spot the bugs in code examples provided for that purpose I can recall common bug types mentioned by my teacher or peers and identify where they might occur in my code	I can debug code with a programming partner I can find a bug once my my teacher or a peer has pointed out its general location I can find my bug after receiving a hint from my teacher or a peer	I can debug my code I can create a bug
	Identify	Develop	Explain
Solve problems by decomposing them into parts	I can identify that algorithms and programming projects can be broken up into parts I can identify the parts of a programming project by running the program and recording what is observable and what it does (reverse engineering)	I can decompose an algorithm into separate parts which could be converted into code	I can explain how decomposition helps me tackle complex problems by breaking up the task into manageable chunks And helping order workflow

	Imitate	*Adapt*	*Create*
Solve problems by decomposing them into parts	I can examine a decomposed programming project or decomposed algorithm and explain the breakdown of tasks	I can finish decomposing a partially completed problem or suggest better ways to decompose a project	I can decompose an algorithm into separate parts which could be converted into code
Pupils should be taught to use sequence, selection, and repetition in programs; work with variables and various forms of input and output			
	Identify	Develop	Explain
Sequence, in programs	I can identify what part of the code is a sequence I can identify which parts of the algorithm will be converted into programming sequences	I can convert an algorithm sequence into a code sequence I can create a code sequence for a purpose	I can explain what is happening in my code sequence to a technical and non-technical person
	Imitate	*Adapt*	*Create*
Sequence in programs	I can follow the steps of a code sequence	I can complete an unfinished code sequence I can reverse a code sequence and explain the differences	I can convert an algorithm sequence into a code sequence I can create a code sequence for a purpose
	Identify	Develop	Explain
Selection in programs	I can identify conditional selection in the code I can identify that selection is about responding to a condition I can identify which parts of the algorithm will be converted into conditional selection code	I can convert an algorithm decision into conditional selection code I can create code that uses conditional selection for a real purpose	I can explain what purpose my conditional selection code fulfils in my program
	Imitate	Adapt	Create
Selection in programs	I can follow an everyday selection example I can read selection code	I can create an everyday selection example I can adapt or reverse a selection example	I can convert an algorithm decision into conditional selection code I can create code that uses conditional selection for a real purpose

	Identify	Develop	Explain
Repetition in programs	I can identify repetition in code I can identify that repetition is about repeating parts of the code I can identify which parts of an algorithm could be converted into repetition code	I can convert an algorithm loop into repetition code I can create code that uses repetition for a real purpose	I can explain what purpose my repetition code fulfils in my program
	Imitate	Adapt	Create
Repetition in programs	I can follow everyday repetition examples I can read repetition code	I can create an everyday repetition example I can adapt or reverse an example of repetition	I can convert repetition in an algorithm into repetition code I can create code that uses repetition for a real purpose
	Identify	Develop	Explain
Work with variables	I can identify variables in code I can identify that variables are about storing, referring to and changing key data within a program I can identify key information in an algorithm that could be converted into code	I can convert key information in an algorithm into variable code I can create code that uses variables for a real purpose	I can explain what purpose my variable fulfils in my program
	Imitate	Adapt	Create
Work with variables	I can think of variables as boxes, pots or storage containers which can have the content changed I can read code with simple variables in it	I can adapt a given variable example (e.g. Change a variable to decrease instead of increase or subtract one variable from another where they previously added)	I can convert key information in an algorithm into variable code I can create code that uses variables for a real purpose

	Identify	Develop	Explain
Work with various forms of input & output	I can identify inputs and outputs in my everyday use of computing I can identify inputs that might be used in a programming context	I can create user input into a program in different ways (e.g. Text, mouse, track pad, touch e.t.c) I can program inputs that are not human (e.g. distance or temperature sensors) I can output my programming in more than one way (e.g. Screen, Sound, physical devices)	I can explain why I chose to use a specific input or output
Pupils should be taught to use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs			
	Identify	Develop	Explain
Use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs	I can identify that algorithms follow logical rules expressed in steps, decisions, loops and variables I can identify that some algorithms & programs don't work because they don't follow logical rules expressed in steps, decisions, loops & variables	I can use my knowledge of the rules that govern algorithms and programs to • Predict how they might operate • Explain how they work • Correct errors	I can explain how reasoning using the rules that govern algorithms and programs helped me • Predict what might happen • Create an algorithm or program • Correct errors
	Imitate	Adapt	Create
Use logical reasoning to detect and correct errors in algorithms or programs	I can look for how my algorithm or code is different from a good example	I can use a checklist of common errors and fixes	I can use my knowledge of algorithmic processes and rules to detect and correct errors in algorithms or programs

Pupils should be taught to understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration			
	Identify	Develop	Explain
Pupils should be taught to understand computer networks including the internet; how they can provide multiple services, such as the world wide web	I can identify that a network is a group of connected people or things I can identify that computer networks are about communicating and collaborating I can identify that the Internet is a network or networks that many home and school devices are connected to	I can explore basic network functions, shared security, work collaboration, communication, storage, printing services etc. I can explore the nature of the internet as a data transport network I can explore some of the main services, such as the World Wide Web, which use the Internet transport network	I can explain in non-technical language how a school or small network works I can explain in non-technical language how the internet connects computers around the world I can explain in non-technical language how multiple services like the world wide web use the internet
	Identify	Develop	Explain
Pupils should be taught to understand computer networks including the Internet and the opportunities they offer for communication and collaboration	I can identify services that people use to communicate and collaborate on a network or the Internet	I can explore ways that a network or Internet communication and collaboration service could be used	I can explain how a network or Internet communication and collaboration service could be used for benefit or for harm
Pupils should be taught to use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content			
	Identify	Develop	Explain
Pupils should be taught to use search technologies effectively	I can identify that the order I add search words affects what results are returned I can identify that there are symbols and keywords that I can use to adapt a search and make it more effective	I can use symbols such as speech marks, tilde ~ , colon:, * asterisk subtract - to refine my search results I can use AND and OR Boolean terms in a search engine effectively	I can explain how either changing the order of my search words, using a symbol or using a Boolean operator improved my search efficiency

	Imitate	Adapt	Create
Pupils should be taught to use search technologies effectively	I can change the order of my search keywords and note the differences in the results returned I can use the keyword kids to return more results aimed at my reading level	I can use speech marks to search for a specific term I can use a subtract symbol before my search to exclude results with that word I can use an asterisk * after an unfinished word to suggest possible endings when searching my files or folders	I can use a tilde ~ to include similar words in my search I can use a colon after keywords to find specific information such as define: author: filetype: intitle: site: I can use Boolean operators AND and OR to find *both* terms or *either* one
	Identify	Develop	Explain
Pupils should be taught to appreciate how results are selected and ranked	I can identify that not all search technologies yield the same results I can identify that I am searching a database of websites not the actual website when I use a search engine I can identify that the top results in a returned search doesn't mean it is the most truthful	I can explore how search results are created and ranked through simplified models	I can explain how search results are selected and ranked
Pupils should be taught to select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information			
	Identify	Develop	Explain
Select	I can identify that some digital tools will be better than others depending on the purpose and intended outcome	I can select the right digital tool that matches the purpose and intended outcome I can choose to use a non-digital tool when it better matches my task and outcome	I can explain why the digital tool I have chosen is right for my project and intended outcome I can explain why I have chosen to use a tool that is not digital

	Identify	Develop	Explain
Combine	I can identify that some tasks will require more than one digital tool, service or device	I can combine digital tools and services to create a project for a specific goal I can combine digital and non-digital tools and services to create a project for a specific goal	I can explain how I combined digital tools or services to create a project for a specific goal I can explain how the use of more than one type of tool or service improved the project outcomes
	Identify	Develop	Explain
Collecting	I can identify that data can be collected using many different types of digital tool	I can use digital systems to collect data I can design digital systems to collect data	I can explain how and why I have collected a specific type of data
	Identify	Develop	Explain
Analysing	I can identify that data can come in many different forms depending on how and why it was collected	I can examine data methodically and in detail looking for patterns and meaning	I can explain the methods and tools I have used to analyse my data I can explain why I am analysing specific data
	Identify	Develop	Explain
Evaluating	I can identify that data can be evaluated to determine how useful it is	I can evaluate data that I have been given or I have collected to determine how useful it is for a specific purpose	I can explain why data is, or isn’t useful for a specific purpose
	Identify	Develop	Explain
Presenting	I can identify that once collected data has been analysed and evaluated it will need to be shared	I can present data that has been analysed and evaluated in a form appropriate for my audience	I can explain why I chose to present data in my chosen way

Pupils should be taught to use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.			
	Identify	Develop	Explain
Pupils should be taught to use technology safely	I can identify examples of safe and unsafe technology use	I can help develop simple safety rules that will keep me safe	I can explain how a specific tool, service or device can be used safely or unsafely
	Identify	Develop	Explain
Pupils should be taught to use technology respectfully	I can identify examples of respectful or disrespectful technology use	I can help develop respectful technology etiquette that will improve my use of technology	I can explain how a specific tool, service, device or interaction can be used or conducted respectfully
	Identify	Develop	Explain
Pupils should be taught to use technology responsibly	I can identify that I am responsible for my use of technology at all times	I can identify that some people may consider that they are not responsible for their actions when using online technology	I can explain how not accepting responsibility for our actions when using technology can harm ourselves and others
	Identify	Develop	Explain
Pupils should be taught to recognise acceptable/unacceptable behaviour	I can recognise acceptable/ unacceptable behaviour	I can recognise my behaviour that is acceptable or unacceptable when using technology	I can explain why my behaviour is acceptable or unacceptable when using technology
	Identify	Develop	Explain
Pupils should be taught to identify a range of ways to report concerns about content and contact.	I can identify many ways that concerns over content or contact can be reported	I know how to report my concerns over content or contact	I can explain why I should report concerns over content and contact

Note

- You could choose **imitate adapt create** and add **explain** on the end
- You would want to adapt these to fit in in with the specific learning context
- Feel free to use and adapt for your school or scheme but please mention the original link and author below
- World Wide Web is often written in capitals but this is not the case in the NC so I have chosen to follow the NC in this document

Draft 2 You can find these on http://code-it.co.uk/afl created by Phil Bagge