

## Transferable skills subject interpretation for the Pearson Edexcel International GCSE in Computer Science (9-1)

Transferable skills will help students cope with the different demands of degree study and provide a solid skills base that enables them to adapt and thrive in different environments across educational stages; and ultimately into employment.

A good international education should enable students to start developing transferable skills as early as possible. Developing these transferable skills where they naturally occur as part of the International GCSE curriculum can help build learner confidence and embed the importance of this well-rounded development. This builds the foundations to ensure students are ready for A-level and higher education.

Our approach to enhancing transferable skills in our International GCSEs ensures that it is not only the academic and cognitive skills that are developed, but those broader elements that universities highlight as being essential for success. Skills such as self-directed study, independent research, self-awareness of own strengths and weaknesses and time-management are skills that students cannot learn from a textbook but have to be developed through the teaching and learning experience that can be provided through an international curriculum.

To support the design of our qualifications, the Pearson Research Team selected and evaluated seven global 21st-century skills frameworks. Following on from this process, we identified the National Research Council's (NRC) framework as the most evidence-based and robust skills framework.

In the tables below, we have taken the NRC framework skills and provided an explicit definition of how each skill can be interpreted for this subject. This will enable teachers and learners to understand examples of how they can develop each skill through this International GCSE.

<b>Intrapersonal skills</b>		<b>Interpersonal skills</b>		<b>Cognitive skills</b>	
Intellectual Openness		Teamwork and collaboration		Cognitive Processes and Strategies	
Adaptability	Persisting in the face of difficulties, such when writing code or designing an algorithm.	Communication	Able to communicate ideas to peers and teachers and to discuss the logic of algorithms and code (verbally or written).	Critical thinking	Clarifying thoughts sufficiently so that they can be expressed in a form that a computer can carry out Analyzing and identifying possibilities and strategies to meet a required process and outcomes. Identifying the outcome of a computational process.
Personal and social responsibility	Appreciating ethical issues in computer science.	Collaboration	Working with peers on shared tasks; giving feedback on peers on problem solving and other tasks.	Problem solving	Solving problems is the core of computer science Problem solving skills include breaking a large problem into a number of smaller ones, recognizing how problems relate to ones that have been solved, setting aside details of a problem that are less important, and identifying and refining the steps needed to reach a solution.
Continuous Learning	Expanding skills and skill-sets through learning and increasing knowledge	Teamwork	Working with peers to solve problems and create programs.	Analysis	Organising ideas and thinking both creatively and logically.
Intellectual interest and curiosity	Willing to tackle challenging problems; researching appropriate solutions and seeking to widen knowledge and increase understanding.	Co-operation	Share ideas with peers and supports peers who are finding tasks difficult.	Reasoning/argumentation	Piecing together information, usually to recommend or suggest further action.
Work ethic/conscientiousness		Interpersonal skills	Giving feedback to peers that is appropriate and delivered in a way that encourages them.	Interpretation	Evaluating alternative solutions or techniques when presented with a small number of well-defined alternatives.
Initiative	Using computational skills to develop own understanding and problem solving abilities.	Leadership		Decision Making	Judging multiple options or alternatives, in order to select one,

Self-direction	Demonstrating autonomous learning.
Responsibility	Taking responsibility for finding and correcting errors in coding and algorithms.
Perseverance	Seeking to remove all errors in code and algorithms using testing and other tools.
Productivity	Using computational skills accurately and efficiently to produce code and algorithms.
Self-regulation (metacognition, forethought, reflection)	Developing an approach to problem solving and program design that maximises both the suitability of the product and the efficiency of the process.
Ethics	Demonstrating awareness of the need to utilise the power of computer science in a way that benefits all.
Integrity	Taking ownership of their own work and responding to challenges.
<b>Positive Core Self Evaluation</b>	
Self-monitoring/self-evaluation/self-reinforcement	Planning and reviewing own work as a matter of course.

Leadership	Leading a group of peers to complete a task.
Responsibility	Taking responsibility for the progress and outcomes of a group task involving problem solving or similar.
Assertive communication	Leading the discussions in a group task ensuring that decisions are made and that group members are all involved.
Self-presentation	Presenting outcomes of a group or individual task to the whole class.

	so as to best fulfil requirements/needs.
Adaptive learning	Using skills, knowledge and understanding to respond the new and unfamiliar challenges.
Executive function	Analysing a situation, planning and taking action, maintaining attention, and adjusting actions as needed to complete a task.
<b>Creativity</b>	
Creativity	Using efficient/effective strategies to create a solution/solve a problem.
Innovation	Suggesting novel solutions to the design/selection of computer science systems.