

Assessment Focuses for Computer Science

	AF 1 – Knowledge and Understanding	AF 2 – Creating Digital Artefacts	AF 3 – Programming	AF 4 – Project Development and Collaboration
8	<ul style="list-style-type: none"> - The student confidently explains complex computing concepts, such as how networks function and the role of protocols. - The student applies advanced knowledge to evaluate the impact of technology on society. - The student demonstrates mastery in computational thinking concepts, such as abstraction and decomposition, across various contexts. 	<ul style="list-style-type: none"> - The student can design complex, fully functional digital products, such as multimedia presentations or websites, incorporating interactivity and clear user interfaces. - The student effectively manages large data sets or inputs and integrates various software tools. 	<ul style="list-style-type: none"> - The student develops complex programs using advanced data structures, like lists or dictionaries, in languages such as Python. - The student can debug and optimize code to improve efficiency and readability. 	<ul style="list-style-type: none"> - The student leads a project with detailed planning, iterative improvements, and thorough evaluation. - The student effectively manages team tasks, resolves conflicts, and integrates diverse ideas.
7	<ul style="list-style-type: none"> - The student can explain how computer systems operate, including key functions like data transmission. - The student can identify technological impacts and demonstrate solid reasoning in analyzing their societal effects. - The student shows a clear understanding of computational thinking, including algorithms and patterns. 	<ul style="list-style-type: none"> - The student can create a functional, user-friendly digital product using various software applications. - The student integrates multimedia elements effectively and adapts to user feedback during design 	<ul style="list-style-type: none"> - The student writes programs using loops and conditionals and can explain how they function. - The student independently debugs code and offers optimized solutions. 	<ul style="list-style-type: none"> - The student follows a structured plan, using feedback to refine and improve the project. - The student contributes to group projects, coordinating efforts and providing technical solutions.
6	<ul style="list-style-type: none"> - The student can describe how key computing systems work, such as the internet and basic protocols. - The student shows an understanding of how technology affects daily life and makes basic comparisons. - The student applies key computational thinking concepts to solve familiar problems. 	<ul style="list-style-type: none"> - The student can create a structured, purposeful digital artifact (e.g., simple website or graphic) using design principles. - The student demonstrates the ability to modify existing digital content to improve functionality or appearance. 	<ul style="list-style-type: none"> - The student writes simple programs incorporating loops and basic data handling (e.g., variables). - The student demonstrates understanding of debugging by fixing common errors. 	<ul style="list-style-type: none"> - The student plans and executes a project with some guidance, demonstrating basic testing and evaluation. - The student works well in teams, contributing ideas and reflecting on collaboration.
5	<ul style="list-style-type: none"> - The student can recall how computers process data and understand simple network concepts. - The student can identify basic impacts of technology on society. - The student can describe algorithms and apply them to simple problem-solving. 	<ul style="list-style-type: none"> - The student can create a basic digital product using standard software, demonstrating understanding of layout and structure. - The student can select appropriate software tools to combine text, images, and data. 	<ul style="list-style-type: none"> - The student can write programs using simple commands and syntax in a block or text-based programming environment. - The student shows awareness of simple debugging methods. 	<ul style="list-style-type: none"> - The student completes a project following a given structure, with basic evaluation and improvements. - The student participates in group work, taking on roles and reflecting on teamwork benefits.
4	<ul style="list-style-type: none"> - The student demonstrates understanding of basic computing terms (e.g., hardware, software) and describes their purpose. - The student recognizes common uses of technology in the real world. - The student begins to explore algorithms, showing limited understanding of basic logic. 	<ul style="list-style-type: none"> - The student can use basic software to create a simple digital artifact (e.g., a poster or a document). - The student understands the need for layout and basic design features (e.g., adding images or headings). 	<ul style="list-style-type: none"> - The student can write simple code using drag-and-drop programming tools, understanding basic logic (e.g., sequence, selection). - The student can edit pre-made code with guidance to achieve specific tasks. 	<ul style="list-style-type: none"> - The student follows a simple project plan, recognizing areas for improvement with guidance. - The student contributes to group discussions and begins to understand teamwork's value.
3	<ul style="list-style-type: none"> - The student can recall some basic computing terminology and identify hardware devices. - The student can describe simple uses of technology in everyday life. - The student can follow simple logical steps in problem-solving using pre-made algorithms. 	<ul style="list-style-type: none"> - The student can use simple commands in basic software (e.g., word processing, painting tools) to create a simple document or drawing. - The student can follow instructions to modify a digital artifact with guidance. 	<ul style="list-style-type: none"> - The student can follow instructions to write a basic program using drag-and-drop coding platforms (e.g., Scratch). - The student understands simple sequences of commands and can modify them slightly. 	<ul style="list-style-type: none"> - The student follows a project template, completing tasks with assistance and identifying areas for improvement. - The student engages in group tasks, completing assigned roles with significant support.
2	<ul style="list-style-type: none"> - The student can identify common computing devices (e.g., mouse, keyboard). The student can recognize basic uses of technology (e.g., typing, drawing). The student can follow very simple instructions to carry out a task on the computer. 	<ul style="list-style-type: none"> - The student can use software with guidance to produce simple outputs, such as typing in a document or drawing using basic tools. 	<ul style="list-style-type: none"> - The student can follow step-by-step instructions to create a very basic program, such as moving an object on screen using a block-based platform. 	<ul style="list-style-type: none"> - The student follows instructions to complete small tasks in a project with substantial guidance. - The student participates in basic group tasks, completing assigned roles with support.