THE ONTARIO CURRICULUM

Digital Technology and Innovations in the Changing World

GRADE 10, OPEN (ICD2O)

2023

Computer Studies

Ontario

The Ontario Public Service endeavours to demonstrate leadership with respect to accessibility in Ontario. Our goal is to ensure that Ontario government services, products, and facilities are accessible to all our employees and to all members of the public we serve. This document, or the information that it contains, is available, on request, in alternative formats. Please forward all requests for alternative formats to ServiceOntario at 1-800-668-9938 (TTY: 1-800-268-7095).

Educators should be aware that, with the exception of **the Grade 10 course Digital Technology and Innovations in the Changing World, 2023 (ICD2O)**, the 2008 computer studies curriculum remains in effect. All secondary computer studies courses for Grades 11–12 will continue to be based on that document. All references to Grade 10 that appear in *The Ontario Curriculum, Grades 10 to 12: Computer Studies, 2008* have been superseded by Digital Technology and Innovations in the Changing World, 2023. As of September 2023, this course replaces Introduction to Computer Studies, Grade 10 (ICS2O), which expired at the end of the 2022–23 school year.

Version history:

Version Date	**Description**
June 14, 2023	New Grade 10 Digital Technology and Innovations in the Changing World, Open course (ICD2O) issued. This course replaces the Grade 10 Introduction to Computer Studies, Open course (ICS2O).

Contents

Une publication équivalente est disponible en français sous le titre suivant : *Technologies numériques et innovations dans un monde en évolution, 10e année, cours ouvert, ICD2O* (2023).

Introduction

Preface

This curriculum policy presents the course Digital Technology and Innovations in the Changing World, Grade 10, Open (ICD2O), 2023. This course supersedes the Grade 10 course, Introduction to Computer Studies, outlined in *The Ontario Curriculum, Grades 10–12: Computer Studies, 2008*. Effective September 2023, all computer studies programs for Grade 10 will be based on the expectations outlined on this site.

In addition to the considerations outlined in this curriculum context, all of the general "Program Planning" sections on this site apply to this course. Educators should review and implement these sections, as well as the components that appear below.

Vision and Goals of the Grade 10 Course, Digital Technology and Innovations in the Changing World

The vision for this course is for students to develop the knowledge and skills related to digital technology and computer programming that will support them in contributing to and leading the global economic, scientific, and societal innovations of tomorrow. Digital technologies play a major part in all aspects of our lives, and the course supports students in understanding how those digital technologies work and how they can be used and developed for the common good. The course enables students to explore how to use computing and critical thinking to address issues that are meaningful to them and their communities, and how to move from being consumers of digital technologies to becoming empowered creators.

Within the course, students are provided with a substantial level of agency and choice in terms of the contexts, intended users, and purposes of the programming projects they undertake. At the same time, students develop knowledge and skills related to digital technologies in their daily lives, including learning related to the use of hardware and software.

As students consider the use and development of digital technologies, they also analyze a broad range of related social, cultural, economic, environmental, and ethical issues. Through this learning, students develop an understanding of the impact of these technologies at a personal, local, and global level.

Throughout the course, students make important connections to other disciplines and careers, and investigate how various industries are changing as a result of digital technology and programming innovations. Their growing understanding of current and emerging digital technologies prepares students for the changing world and supports them in potentially becoming the digital leaders of tomorrow.

The goals of the Grade 10 course, Digital Technology and Innovations in the Changing World, are to enable all students to:

- gain an understanding of how digital technologies work, including how they are designed, implemented, and used in society;
- develop the computational thinking skills needed to design computer programs;
- develop the foundational programming skills required to read, understand, and write programs;
- gain perspectives and insights related to social, cultural, economic, environmental, and ethical issues that will support them in making responsible choices with respect to digital technologies;
- develop lifelong learning skills and insights related to digital technology and innovations that will help them adapt and thrive in the changing world, including the workplace;
- make connections that will help them better understand digital technology and its influence and impact on their lives and various communities;
- investigate various career options related to digital technology and programming; and
- feel empowered to pursue further studies in computer technology and/or computer science.

The Power of Digital Technology and Computer Programming

This course provides all students with opportunities to appreciate, explore, and discover the power of digital technology and computer programming.

Students live in a world where digital technologies connect us, entertain us, and protect us. These technologies have become powerful aspects of our lives. As such, it is important for students to appreciate their potential benefits and also to think critically about their drawbacks, such as excessive or problematic use. While this course provides students with opportunities to examine important social, cultural, economic, environmental, and ethical issues related to digital technology and computer programming, they are also exposed to the excitement and promise of current, emerging, and future innovations. Students also develop their appreciation of the critical roles of human creativity, empathy, and ethics in digital technology innovations and solutions.

In this course, students are provided with hands-on opportunities to explore the process of designing and creating computational artifacts. Student engagement and inspiration are stimulated as they develop their projects, and students build pride in their accomplishments as they share their creations.

Industries and occupations are continually changing as a result of digital technology innovation. As students enhance their digital technology and programming skills, they are encouraged to consider how they could apply these skills to develop programs that could contribute to the innovations of tomorrow.

The Importance of Computer Studies in STEM Education

STEM education is the cross-curricular study of science, technology, engineering, and mathematics (STEM), and the application of those subjects in real-world contexts. As students engage in STEM education, they develop the transferable skills that they need to meet the demands of today's global economy and society and to become digitally literate citizens.

Skills developed through STEM education include those related to computational thinking and programming. These skills are in high demand in today's changing world, as digital technologies continue to impact all areas of our lives. In this course, students use computational thinking, design, and programming skills to create computational artifacts. They also connect these skills to other subject areas, and to various careers, including skilled trades, recognizing that digital technology and computer programming have close connections to many areas of STEM.

The integration of knowledge and skills from a number of other STEM-related disciplines into this course can reinforce students' understanding of each of these disciplines and of the interrelationships among them. Similarly, reflecting on diverse perspectives engages students in a variety of creative and critical-thinking processes that are essential for developing innovative, ethical, and effective responses to various issues.

The weaving of themes and components of STEM education throughout this course is intended to foster creativity, critical thinking, and problem solving. This course also supports the development of digital citizenship in students, enabling them to better appreciate, understand, and responsibly navigate the digital world in which they live.

Elements of the Grade 10 Course, Digital Technologies and Innovations in the Changing World

Overview

This course is designed to be inclusive of all students, to provide them with opportunities to create programs that are relevant and responsive to their needs and interests, and to enable them to extend their learning. The course acknowledges and builds on the coding concepts and skills students have learned in earlier grades. The focus of the course is on consolidating past learning and deepening students' understanding of these concepts and skills.

Throughout this course, students apply computational thinking concepts and practices to develop programs for a wide variety of contexts, users, and purposes. Students develop an understanding of important issues, contributions, and innovations related to digital technology. They investigate applications of digital technology skills and programming concepts and skills, and make connections to other fields and to potential future careers.

Students expand their knowledge of hardware devices and software applications, including those that they use every day. Students investigate innovations that impact their daily lives, such as those related to artificial intelligence, data collection, automation, networking, and cybersecurity. This course also provides opportunities for students to investigate concepts and practices related to cyber safety and digital citizenship, important considerations for students in an increasingly connected world. The course helps students understand the role and importance of computer science and digital technology in all fields, and enables them to develop fundamental programming knowledge and skills that they can apply in other computer studies or STEM-related courses.

The course information that appears in the next section is in effect starting in the 2023–24 school year. The 2008 computer studies curriculum for Grades 11 and 12 remains in effect. All references to Grade 10 that appear in *The Ontario Curriculum, Grades 10 to 12: Computer Studies, 2008* have been superseded by the section below.

Courses in Computer Studies, Grades 10 to 12

The computer studies program comprises courses in Grades 10, 11, and 12.

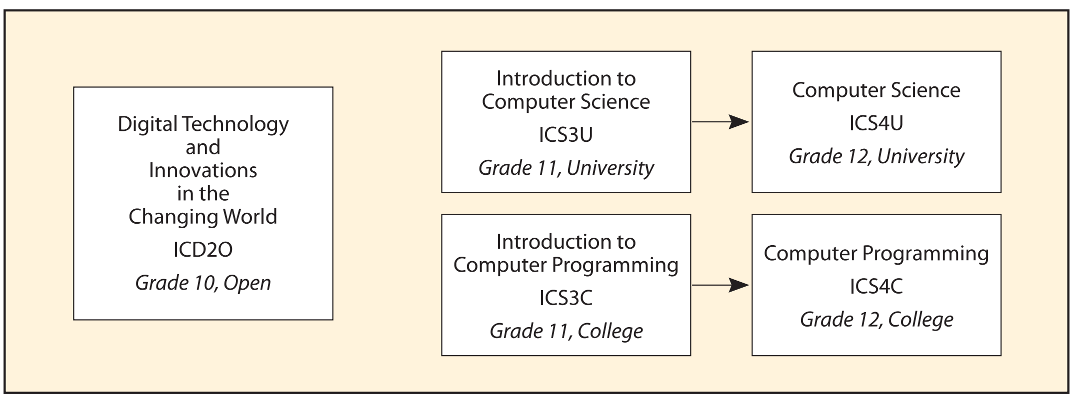
The Grade 10 course, Digital Technologies and Innovations in the Changing World, is designated as an open course. Open courses are designed to broaden students' knowledge and skills in subjects that reflect their interests and to prepare them for active and rewarding participation in society. They are not designed with the specific requirements of universities, colleges, or the workplace in mind.

Grade	Course Name	Course Type	Course Code	Prerequisite
10	Digital Technology and Innovations in the Changing World	Open	ICD2O	None
11	Introduction to Computer Science	University	ICS3U	None
11	Introduction to Computer Programming	College	ICS3C	None
12	Computer Science	University	ICS4U	Grade 11 Introduction to Computer Science, University
12	Computer Programming	College	ICS4C	Grade 11 Introduction to Computer Programming, College

Note: Each of the courses listed above is worth one credit.

Prerequisite Chart for Computer Studies, Grades 10–12

This chart maps out all the courses in computer studies and shows the links between courses and the possible prerequisites for them.



Although courses in computer studies are optional, students should keep in mind that any computer studies course in the Grade 10–12 program can fulfil an additional Group 3 compulsory credit requirement for the Ontario Secondary School Diploma.

Half-Credit Courses

The course outlined in this curriculum is designed to be offered as a full-credit course. However, it may also be delivered as a half-credit course. Half-credit courses, which require a minimum of fifty-five hours of scheduled instructional time, must adhere to the following conditions:

- The two half-credit courses created from a full course must together contain all of the expectations of the full course. Students must successfully complete both parts of the course if it is to be used as a prerequisite for another course.
- The title of each half-credit course must include the designation Part 1 or Part 2. A half credit (0.5) will be recorded in the credit-value column of both the report card and the Ontario Student Transcript.

Boards will report all half-credit courses to the ministry annually in the School October Report.

Curriculum Expectations

The expectations identified for this course describe the skills and knowledge that students are expected to acquire, demonstrate, and apply in their class work and tasks, on tests, in demonstrations, and in various other activities on which their achievement is assessed and evaluated.

Mandatory learning is described in the overall and specific expectations of the curriculum.

Two sets of expectations – overall expectations and specific expectations – are listed for each *strand*, or broad area of the curriculum. The strands in this course are lettered A through C. *Taken together, the overall and specific expectations represent the mandated curriculum.*

The *overall expectations* describe in general terms the skills and knowledge that students are expected to demonstrate by the end of the course. The *specific expectations* describe the expected skills and knowledge in greater detail. The specific expectations are organized under numbered subheadings, each of which indicates the strand and the overall expectation to which the group of specific expectations corresponds (e.g., “B2” indicates that the group relates to overall expectation 2 in Strand B). This organization is not meant to imply that the expectations in any one group are achieved independently of the expectations in the other groups, nor is it intended to imply that the learning associated with the expectations happens in a linear, sequential way. The numbered headings are used merely as an organizational structure to help teachers focus on particular aspects of knowledge, concepts, and skills as they develop various lessons and learning activities for students.

Teacher Supports

Specific expectations are accompanied by supports such as examples, teacher prompts, and/or instructional tips. [1] Examples are meant to clarify the requirement specified in the expectation, illustrating the kind of skill or knowledge, the specific area of learning, the depth of learning, and/or the level of complexity that the expectation entails. Teacher prompts are sample guiding questions and considerations that can lead to discussions and promote deeper understanding. Instructional tips suggest instructional strategies and authentic contexts for the effective modelling, practice, and application of computer studies concepts.

Teacher supports, such as the examples, teacher prompts, and instructional tips, are optional supports that educators can draw on to support teaching and learning, in addition to developing their own supports that reflect a similar level of complexity. Whatever the specific ways in which the requirements outlined in the expectations are implemented in the classroom, they must be inclusive and, wherever possible, reflect the diversity of the student population.

Computational Thinking

Computational thinking is a model of thinking that is more about thinking than it is about computing. It is about designing and evaluating potential solutions to problems – often through programming. The concepts of computational thinking include:

- decomposition (the breakdown of a problem or task into steps or pieces)
- pattern recognition (the identification of other problems or items that are similar)
- abstraction (the reduction of a complex task to its essential components)
- algorithms (a set of instructions to follow to solve a problem)

When these concepts are applied, they are known as computational thinking practices.

In this course, the computational thinking concepts and practices are situated in Strand A and applied throughout the learning in Strands B and C. For example, students consider how abstraction is used in the design of computers and hardware devices they use every day. Students also use abstraction and pattern recognition to inform the design of algorithms. In addition, students break up larger tasks (decomposition) and work iteratively, solving smaller steps that contribute to the larger solution.

Although computational thinking is considered a component of computer science, students are encouraged to consider how it can be applied in other disciplines and careers, including skilled trades.

[1] The teacher supports will be made available at a later date, after the issuing of the curriculum expectations and the curriculum context.

The Strands in the Grade 10 Course, Digital Technology and Innovations in the Changing World

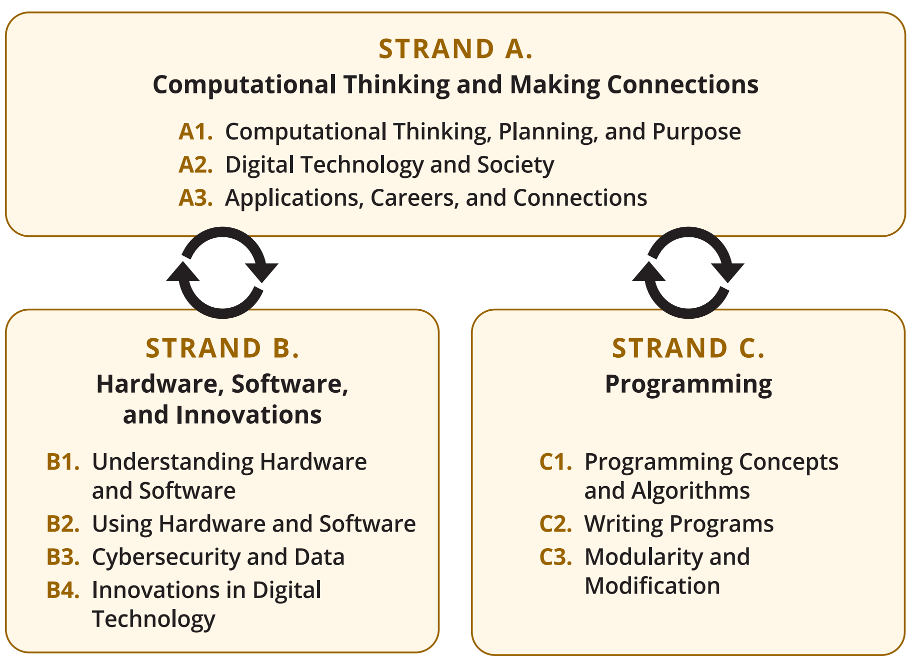
The expectations in the course are organized into three distinct but related strands. Strand A is an overarching strand focusing on the skills and considerations that will enable students to investigate concepts and integrate knowledge from the other two strands, and to make connections between computer studies and other disciplines. This strand also encourages students to examine various careers, including those in the skilled trades. In Strands B and C, students integrate Strand A expectations as they develop their understanding of strand-specific concepts; investigate hardware, software, and innovations; and use the concepts of programming and algorithms to design and create programs.

Throughout the course, learning related to the expectations in Strand A occurs in the context of learning related to the other two strands.

The three strands are as follows:

- A. Computational Thinking and Making Connections
- B. Hardware, Software, and Innovations
- C. Programming

The chart below illustrates the relationship between Strand A and the other two strands:



Strand A – Computational Thinking and Making Connections

Strand A develops students' understanding of computational thinking concepts and practices, and their use in designing algorithms and developing programs that support the needs of a variety of users.

In this strand, students analyze a variety of societal issues related to digital technology, taking various perspectives into account. They apply critical thinking skills to investigate the benefits and limitations of digital technologies. Students explore the relevance of programming and the impacts of digital technology innovations and cybersecurity issues on their daily lives and the lives of others. They also consider the importance of accessibility in relation to digital technology.

Students make connections between what they are learning about digital technology and programming and the application of this learning in different disciplines and industries, as well as in skilled trades.

Strand B – Hardware, Software, and Innovations

In Strand B, students develop an understanding of modern hardware devices and software applications, and how they can meet the needs of various users.

In this strand, students investigate how data and connectivity are integral components of many of the applications and devices they use every day. While learning about the collection and management of data in various contexts, students also explore and apply safe and effective cybersecurity practices. In addition, students investigate current and emerging innovations, including artificial intelligence, and their impact on everyday life, both today and in the future.

Strand C – Programming

In this strand, students build on their coding experiences in previous courses and grades. Students use the computational thinking practices as a framework for problem solving as they design algorithms and write programs using various programming concepts. They also develop the skills to interpret program errors and document their programs to enable collaboration with others.

Program Planning and Cross-Curricular and Integrated Learning

Educators consider many factors when planning a computer studies program that cultivates the best possible environment in which all students can maximize their learning. This section highlights important areas of focus that educators should consider, including areas of cross-curricular and integrated learning, as they plan effective and accessible computer studies programs. In addition, all of the general "Program Planning" sections on this site apply to this curriculum.

Instructional Approaches in the Grade 10 Course, Digital Technology and Innovations in the Changing World

Instruction in Grade 10 computer studies should support all students in acquiring the knowledge, skills, and habits of mind that they need in order to: investigate issues and innovations; apply computational thinking concepts and practices; design and share algorithms and programs; think critically; and work cooperatively on exciting projects.

Universal Design for Learning (UDL) and Differentiated Instruction (DI)

Students in every computer studies classroom vary in their identities, lived experiences, personal interests, learning profiles, and readiness to learn new concepts and skills. Universal Design for Learning (UDL) and differentiated instruction (DI) are robust and powerful approaches to designing assessment and instruction to engage all students in computer studies tasks that develop conceptual understanding. UDL and DI can be used in combination to help teachers respond effectively to the strengths and needs of all students.

The aim of the UDL framework is to assist teachers in designing computer studies programs and environments that provide all students with equitable access to the computer studies curriculum. Teachers take into account students' diverse learner profiles by designing tasks that offer individual choice, ensuring relevance and authenticity, providing graduated levels of challenge, and fostering collaboration in the computer studies classroom. Teachers also represent concepts and information in multiple ways to help students become resourceful and knowledgeable learners. For example, teachers use a variety of media to ensure that students are provided with alternatives for auditory and visual information. To support learners as they focus strategically on their learning goals, teachers create an environment in which learners can express themselves using a range of kinesthetic, visual, and auditory strengths. For example, teachers can vary ways in which students can respond and demonstrate their understanding of concepts, and can support students in goal-setting, planning, and time-management skills related to their computer studies learning.

Designing programming assignments and tasks through UDL allows the learning to be “low floor, high ceiling” – that is, all students are provided with the opportunity to find their own entry point to the learning. Teachers can then support students in working at their own pace and can provide further support as needed, while continuing to move student learning forward. Programming and design tasks that are intentionally created to be low floor, high ceiling provide opportunities for students to use varied approaches and to continue to be engaged in learning with varied levels of complexity and challenge. This is an inclusive approach that is grounded in a growth mindset: the belief that everyone can do well in computer studies.

While UDL provides teachers with broad principles for planning computer studies instruction and learning experiences for a diverse group of students, DI allows them to address specific skills and learning needs. DI is student centred and involves a strategic blend of whole-class, small-group, and individual learning activities to suit students’ differing strengths, interests, and levels of readiness to learn. Attending to students’ varied readiness for learning in computer studies is an important aspect of differentiated teaching. For example, learners who are ready for greater challenges need support in aiming higher, developing belief in excellence, and co-creating problem-based tasks of increasing complexity while still maintaining joy in learning. At the same time, students who are struggling to learn a concept need to be provided with the scaffolding and encouragement to reach high standards. To make certain concepts more accessible, teachers can employ strategies such as offering students choice and providing open-ended problems that are based on relevant real-life situations and supported with visual and hands-on learning.

Universal Design for Learning and differentiated instruction are integral aspects of an inclusive computer studies program with the goal of achieving equity in computer science education. More information on these approaches can be found in the ministry publication *Learning for All: A Guide to Effective Assessment and Instruction for All Students, Kindergarten to Grade 12* (2013).

Additional Instructional Approaches for Computer Science

Students learn best when they are engaged in learning in a variety of ways. The concepts and skills included in this course can be taught within a number of different contexts that resonate with students and that connect to their lives, their interests, their future goals, and the digital technologies they use. By varying instructional approaches in computer programming and digital technology in ways that address individual needs and interests, teachers can encourage all students to see themselves as capable learners.

When students are engaged in practical, hands-on, and experiential learning, they can develop their understanding of the concepts in the course, make connections between these concepts and their practical application, and practise and refine their skills. As concepts are introduced, teachers are encouraged to consider using an active learning approach, such as live coding. *Live coding* is a demonstration by the teacher in which they explain each step of the problem-solving and programming processes as students engage with these processes in real time. Teachers can deliberately introduce errors in order to demonstrate how to respond to such difficulties. This approach provides opportunities for students to consolidate their understanding and develop their debugging and programming skills.

Teachers are encouraged to pace live coding activities with care to ensure that all students can actively participate and have time to formulate and ask questions to clarify their understanding.

Teachers are also encouraged to scaffold activities and projects – for example, guiding students as they move from reading and using pre-existing code, to modifying components of an already written program, and to developing their own code and programs. A commonly used scaffolding technique involves the use of worked examples. A *worked example* could be a partially completed piece of code, including explanatory annotations, showing a possible solution to a frequently encountered problem. Worked examples can support learning and build students' confidence, enabling them to approach more challenging problems.

Students can benefit from working individually to investigate algorithms and write software programs, and they can also benefit from collaborative work. Pair programming and peer instruction are two examples of frequently used collaborative learning strategies in computer studies.

Pair programming is a technique in which two students – a driver and a navigator – work together using a single computer to solve a problem. The driver's role is to write the code, while the navigator provides advice and guidance as they jointly work towards achieving a common goal.

Peer instruction involves the use of targeted multiple-choice questions with distractors that are designed to expose possible misconceptions. It can be an effective technique to check for understanding and to encourage student dialogue about course topics. The peer instruction process involves the following steps:

1. Students investigate or practise using new concepts.
2. The teacher poses a multiple-choice question, and students individually select their answers.
3. Students discuss their choices with their peers, which enables them to explore the topic and possibly clarify their understanding.
4. The teacher poses the same question again and asks each student to reassess their answer.
5. The teacher facilitates a whole-group discussion of the topic under consideration.

Teachers are also encouraged to provide students with opportunities to celebrate their success through program showcases or demonstrations, which can serve as an exciting culminating activity in the course.

Coding: Consolidating and Expanding Prior Learning

Digital Technology and Innovations in the Changing World is the earliest dedicated course in computer studies, yet students enter this course having had a variety of earlier experiences with coding concepts and skills in Grades 1 to 8 mathematics, Grade 9 mathematics, Grades 1 to 8 science and technology, and Grade 9 science. Teachers can encourage students to draw on and consolidate their prior knowledge and skills throughout this course.

Students may have used various programming languages and environments in a variety of contexts in earlier grades to support their learning of concepts and skills in mathematics and in science and technology. This variety should be seen as an asset, permitting rich discussions and sharing of prior learning and experiences. Teachers may find it worthwhile to become familiar with students' earlier work and experiences with coding, to have them share and build upon those experiences, and to have past work serve as seeds for project ideas and innovations in this course.

Students' prior experience with coding may have included block-based coding environments. Teachers should initially focus on linking fundamental coding concepts as students explore the extended learning opportunities provided in Strand C of this course. It is important that teachers ensure that students follow consistent programming practices and conventions, as they learn appropriate terminology to describe programming concepts, constructs, and algorithms, and as they refine their skills. Such an approach will serve all students, but particularly those who decide to further develop their computer studies skills in senior STEM-related courses.

Innovations and Emerging Technology

A central focus of this course is learning related to innovations and emerging technologies, including the social, cultural, economic, environmental, and ethical issues related to their development and use. These can be engaging topics that capture the imagination of students as they consider exciting innovations in digital technologies and imagine themselves playing a role in the development and application of these innovations, contributing to a hopeful and exciting future.

These topics also provide students with opportunities to critically assess technologies and to consider issues surrounding digital accessibility, privacy, appropriate use, bias, ethical design, and environmental sustainability.

Students also analyze contributions to, and innovations in, digital technology by people from diverse local, Canadian, and global communities, including Indigenous communities in Canada and around the world. As students engage with this learning, they are empowered to consider that they can help shape the future in a positive way, potentially contributing to the development of future innovations by pursuing careers or further education in computer science or other STEM-related areas, including skilled trades.

Skilled Trades

A *skilled trade* is a career path that requires hands-on work and specialty knowledge. Many skilled trades workers apply STEM-related concepts as they construct buildings; build and maintain infrastructure for transportation, communications, and utilities; or provide a range of professional services. Automation, digital technology, and computer programming have a great impact on these sectors, making components of this course especially relevant to students pursuing skilled trades pathways and careers.

Throughout this course, students will investigate how digital technology and programming concepts and skills can be used in other disciplines and in real-world applications. Students also explore ways in which various industries are changing as a result of digital technology and programming innovations. Career-related expectations provide opportunities for students to connect concepts and skills associated with this course to potential postsecondary education and career pathways, including in the skilled trades. As students discover the powerful relationships between hardware and software, they can be encouraged to consider further developing their interfacing and digital electronics skills in the computer technology courses in the senior grades of the technological education curriculum.

Teachers are encouraged to provide valuable experiential learning opportunities that connect students with role models with diverse lived experiences. An excellent opportunity to do so may include classroom presentations given by guest speakers from populations that are underrepresented in the skilled trades, such as women engaged in technical trades that rely on digital technology.

Financial Literacy

Financial literacy education provides students with the preparation they need to make informed decisions in a complex and fast-changing financial world. This course provides a number of opportunities for students to develop skills and knowledge related to financial literacy.

To become responsible digital citizens in a global economy, students need to develop their understanding of the implications of their own choices as consumers. In this course, students have opportunities to consider societal and environmental issues associated with the purchase, use, and disposal of digital technologies. They learn to take budgetary constraints into account as they assess user needs and identify the hardware and software that would be appropriate to specific situations. Students explore the financial impacts associated with data breaches that can occur within private and public institutions, and they develop their understanding of the importance of responsibly managing data and mitigating risk.

In this course, students develop their critical thinking skills around financial literacy as they explore the economic impacts of digital technology innovations on various industries and occupations. Students also identify measures and technologies that promote an environmentally and economically sustainable digital future.

Assessment and Evaluation of Student Achievement

Growing Success: Assessment, Evaluation, and Reporting in Ontario Schools, First Edition, Covering Grades 1 to 12, 2010 sets out the Ministry of Education's assessment, evaluation, and reporting policy. The policy aims to maintain high standards, improve student learning, and benefit all students, parents[2], and teachers in elementary and secondary schools across the province. Successful implementation of this policy depends on the professional judgement[3] of teachers at all levels as well as their high expectations of all students, and on their ability to work together and to build trust and confidence among parents and students.

Major aspects of assessment, evaluation, and reporting policy are summarized in the general "Assessment and Evaluation" section that applies to all curricula. The key tool for assessment and evaluation in the Grade 10 course, Digital Technology and Innovations in the Changing World – the achievement chart – is provided below.

The Achievement Chart for the Grade 10 Computer Studies Course

The achievement chart identifies four categories of knowledge and skills and four levels of achievement in the Grade 10 course, Digital Technologies and Innovations in the Changing World. (For important background, see "Content Standards and Performance Standards" in the general "Assessment and Evaluation" section that applies to all curricula.)

[2] The word *parent(s)* is used on this website to refer to parent(s) and guardian(s). It may also be taken to include caregivers or close family members who are responsible for raising the child.

[3] "Professional judgement", as defined in *Growing Success* (p. 152), is "judgement that is informed by professional knowledge of curriculum expectations, context, evidence of learning, methods of instruction and assessment, and the criteria and standards that indicate success in student learning. In professional practice, judgement involves a purposeful and systematic thinking process that evolves in terms of accuracy and insight with ongoing reflection and self-correction".

Knowledge and Understanding – Subject-specific content acquired in this course (knowledge), and the comprehension of its meaning and significance (understanding)				
Categories	**50–59% (Level 1)**	**60–69% (Level 2)**	**70–79% (Level 3)**	**80–100% (Level 4)**
	The student:			
Knowledge of content *(e.g., facts, technical terminology, computational thinking concepts*, programming concepts)*	demonstrates limited knowledge of content	demonstrates some knowledge of content	demonstrates considerable knowledge of content	demonstrates thorough knowledge of content
Understanding of content *(e.g., processes, concepts, tools, computational thinking practices*)*	demonstrates limited understanding of content	demonstrates some understanding of content	demonstrates considerable understanding of content	demonstrates thorough understanding of content
Thinking – The use of critical and creative thinking skills and/or processes				
Categories	**50–59% (Level 1)**	**60–69% (Level 2)**	**70–79% (Level 3)**	**80–100% (Level 4)**
	The student:			
Use of planning skills *(e.g., identifying a need or problem, gathering information, selecting strategies and tools, setting goals, developing timelines)*	uses planning skills with limited effectiveness	uses planning skills with some effectiveness	uses planning skills with considerable effectiveness	uses planning skills with a high degree of effectiveness
Use of processing skills *(e.g., analyzing a need or problem, carrying out a plan to create programs)*	uses processing skills with limited effectiveness	uses processing skills with some effectiveness	uses processing skills with considerable effectiveness	uses processing skills with a high degree of effectiveness
Use of critical/creative thinking processes *(e.g., problem solving, research and inquiry)*	uses critical/ creative thinking processes with limited effectiveness	uses critical/ creative thinking processes with some effectiveness	uses critical/ creative thinking processes with considerable effectiveness	uses critical/ creative thinking processes with a high degree of effectiveness

Communication – The conveying of meaning through various forms				
Categories	**50–59% (Level 1)**	**60–69% (Level 2)**	**70–79% (Level 3)**	**80–100% (Level 4)**
	The student:			
Expression and organization of ideas and information *(e.g., clear expression, logical organization)* **in oral, visual, and/or written forms** *(e.g., demonstrations, interviews, presentations, reports, flowcharts, pseudocode, code)*	expresses and organizes ideas and information with limited effectiveness	expresses and organizes ideas and information with some effectiveness	expresses and organizes ideas and information with considerable effectiveness	expresses and organizes ideas and information with a high degree of effectiveness
Communication for different audiences *(e.g., peers, users)* **and purposes** *(e.g., to inform, to persuade, to solve problems, to collaborate)* **in oral, visual, and/or written forms**	communicates for different audiences and purposes with limited effectiveness	communicates for different audiences and purposes with some effectiveness	communicates for different audiences and purposes with considerable effectiveness	communicates for different audiences and purposes with a high degree of effectiveness
Use of conventions, vocabulary, and terminology of the discipline in oral, visual, and/or written forms *(e.g., terms, programming language syntax, coding standards)*	uses conventions, vocabulary, and terminology with limited effectiveness	uses conventions, vocabulary, and terminology with some effectiveness	uses conventions, vocabulary, and terminology with considerable effectiveness	uses conventions, vocabulary, and terminology with a high degree of effectiveness

Application – The use of knowledge and skills to make connections within and between various contexts				
Categories	**50–59% (Level 1)**	**60–69% (Level 2)**	**70–79% (Level 3)**	**80–100% (Level 4)**
	The student:			
Application of knowledge and skills *(e.g., tools, computational thinking concepts, programming concepts and strategies)* **in familiar contexts**	applies knowledge and skills in familiar contexts with limited effectiveness	applies knowledge and skills in familiar contexts with some effectiveness	applies knowledge and skills in familiar contexts with considerable effectiveness	applies knowledge and skills in familiar contexts with a high degree of effectiveness
Transfer of knowledge and skills *(e.g., tools, computational thinking concepts, programming concepts and strategies)* **to new contexts**	transfers knowledge and skills to new contexts with limited effectiveness	transfers knowledge and skills to new contexts with some effectiveness	transfers knowledge and skills to new contexts with considerable effectiveness	transfers knowledge and skills to new contexts with a high degree of effectiveness
Making connections within and between various contexts (*e.g., connections to everyday personal situations; connections to social, cultural, economic, environmental, and ethical issues; connections between computer studies and other STEM [science, technology, engineering, and mathematics] subjects; connections to potential careers)*	makes connections within and between various contexts with limited effectiveness	makes connections within and between various contexts with some effectiveness	makes connections within and between various contexts with considerable effectiveness	makes connections within and between various contexts with a high degree of effectiveness

* See the section "Computational Thinking" for more information.

Criteria and Descriptors for Grade 10 Digital Technology and Innovations in the Changing World

To guide teachers in their assessment and evaluation of student learning, the achievement chart provides "criteria" and "descriptors" within each of the four categories of knowledge and skills.

A set of criteria is identified for each category in the achievement chart. The criteria are subsets of the knowledge and skills that define the category. The criteria identify the aspects of student performance that are assessed and/or evaluated, and they serve as a guide to what teachers look for. In the Grade 10 computer studies course, the criteria for each category are as follows:

Knowledge and Understanding

- knowledge of content (e.g., facts, technical terminology, computational thinking concepts, programming concepts)
- understanding of content (e.g., processes, concepts, tools, computational thinking practices)

Thinking

- use of planning skills (e.g., identifying a need or problem, gathering information, selecting strategies and tools, setting goals, developing timelines)
- use of processing skills (e.g., analyzing a need or problem, carrying out a plan to create programs)
- use of critical/creative thinking processes (e.g., problem solving, research and inquiry)

Communication

- expression and organization of ideas and information (e.g., clear expression, logical organization) in oral, visual, and/or written forms (e.g., demonstrations, interviews, presentations, reports, flowcharts, pseudocode, code)
- communication for different audiences (e.g., peers, users) and purposes (e.g., to inform, to persuade, to solve problems, to collaborate) in oral, visual, and/or written forms
- use of conventions, vocabulary, and terminology of the discipline in oral, visual, and/or written forms (e.g., terms, programming language syntax, coding standards)

Application

- application of knowledge and skills (e.g., tools, computational thinking concepts, programming concepts and strategies) in familiar contexts
- transfer of knowledge and skills (e.g., tools, computational thinking concepts, programming concepts and strategies) to new contexts
- making connections within and between various contexts (e.g., connections to everyday personal situations; connections to social, cultural, economic, environmental, and ethical issues; connections between computer studies and other STEM [science, technology, engineering, and mathematics] subjects; connections to potential careers)

Descriptors indicate the characteristics of the student's performance, with respect to a particular criterion, on which assessment or evaluation is focused. *Effectiveness* is the descriptor used for each criterion in the Thinking and Investigation, Communication, and Application categories. What constitutes effectiveness in any given performance task will vary with the particular criterion being considered. Assessment of effectiveness may therefore focus on a quality such as appropriateness, clarity, accuracy, precision, logic, relevance, significance, fluency, flexibility, depth, or breadth, as appropriate for the particular criterion.

Expectations by Strand

A. Computational Thinking and Making Connections

This strand focuses on students learning how to apply the transferable skills in the context of computer studies and developing skills associated with computational thinking. Students will investigate a wide range of issues related to digital technology, and will also make connections between computer studies and digital citizenship while considering various perspectives. The learning related to this strand takes place in the context of learning related to Strand B: Hardware, Software, and Innovations, and Strand C: Programming, and it should be assessed and evaluated within these contexts.

Overall expectations

Throughout this course, in connection with the learning in the other strands, students will:

A1. Computational Thinking, Planning, and Purpose

apply computational thinking concepts and practices, and use various tools[4] and processes to plan and develop computational artifacts for a wide variety of contexts, users, and purposes

Specific expectations

Throughout this course, in connection with the learning in the other strands, students will:

A1.1 apply computational thinking concepts and practices when planning and designing computational artifacts

A1.2 use a variety of tools and processes to plan, design, and share algorithms and computational artifacts

A1.3 develop computational artifacts for a variety of contexts and purposes that support the needs of diverse users and audiences

---

[4] Planning tools may include flowcharts, pseudocode, and so on. Development tools may include code editors, integrated development environments (IDEs), and so on.

A2. Digital Technology and Society

demonstrate an understanding of important social, cultural, economic, environmental, and ethical issues, as well as contributions and innovations involving diverse local and global communities, related to digital technology

Specific expectations

Throughout this course, in connection with the learning in the other strands, students will:

A2.1 investigate current social, cultural, economic, environmental, and ethical issues related to digital technology that have personal, local, and global impacts, taking various perspectives into account

A2.2 analyze personal and societal safety and cybersecurity issues related to digital technology, and identify measures and technologies that can help mitigate related concerns for individuals and communities

A2.3 investigate contributions to innovations in digital technology and computing by people from diverse local, Canadian, and global communities, including Indigenous communities in Canada and around the world

A2.4 investigate how to identify and address bias involving digital technology

A2.5 analyze accessibility issues involving digital technology, and identify measures that can improve accessibility

A3. Applications, Careers, and Connections

demonstrate an understanding of real-world applications of digital technology and programming, including within various industries and careers

Specific expectations

Throughout this course, in connection with the learning in the other strands, students will:

A3.1 investigate how digital technology and programming skills can be used within a variety of disciplines in real-world applications

A3.2 investigate ways in which various industries, including those that involve skilled trades, are changing as a result of digital technology and programming innovations

A3.3 investigate various career options related to digital technology and programming, and ways to continue their learning in these areas

B. Hardware, Software, and Innovations

Overall expectations

By the end of this course, students will:

B1. Understanding Hardware and Software

demonstrate an understanding of the functions and features[5] of the hardware and software they encounter in their everyday life

Specific expectations

By the end of this course, students will:

B1.1 describe the functions and features of various core components of hardware associated with digital technologies they encounter in their everyday life

B1.2 describe the functions and features of various connected devices associated with digital technologies they encounter in their everyday life

B1.3 describe the functions of various types of software they encounter in their everyday life

B2. Using Hardware and Software

demonstrate an understanding of various ways to use hardware, software, and file management, and of research practices to support their own use of digital technology

Specific expectations

By the end of this course, students will:

B2.1 use file management techniques, including those related to local and cloud storage, to organize, edit, and share files

B2.2 identify and use effective research practices and supports when learning to use new hardware or software

B2.3 assess the hardware and software requirements for various users, contexts, and purposes in order to make recommendations for devices and programs

---

[5] "Functions" refers to the purpose of the components, devices, or software. "Features" refers to units of measure, such as the transfer rates (bits/second) and capacity limits (bytes) of memory and storage components.

B3. Cybersecurity and Data

demonstrate an understanding of safe and effective practices related to data and cybersecurity in various contexts

Specific expectations

By the end of this course, students will:

B3.1 apply safe and effective data practices when using digital technology in various contexts

B3.2 apply safe and effective security practices, including practices to protect their privacy, when using digital technology in various contexts

B4. Innovations in Digital Technology

investigate current and emerging innovations in digital technology, including automation and artificial intelligence, and assess their benefits and limitations

Specific expectations

By the end of this course, students will:

B4.1 investigate current innovations, including automation and artificial intelligence systems, and assess the impacts of these technologies on everyday life

B4.2 investigate hardware and methods used to establish networks and connectivity, and assess the benefits and limitations of increased connectivity with reference to everyday life

B4.3 investigate emerging innovations related to hardware and software and their possible benefits and limitations with reference to everyday life in the future

C. Programming

Overall expectations

By the end of this course, students will:

C1. Programming Concepts and Algorithms

explain fundamental programming concepts and algorithms

Specific expectations

By the end of this course, students will:

C1.1 use appropriate terminology to describe programming concepts and algorithms

C1.2 describe simple algorithms that are encountered in everyday situations

C1.3 identify various types of data and explain how they are used within programs

C1.4 determine the appropriate expressions and instructions to use in a programming statement, taking into account the order of operations

C1.5 identify and explain situations in which conditional and repeating structures are required

C2. Writing Programs

use fundamental programming concepts to write simple programs

Specific expectations

By the end of this course, students will:

C2.1 use variables, constants, expressions, and assignment statements to store and manipulate numbers and text in a program

C2.2 write programs that use and generate data involving various sources and formats

C2.3 write programs that include single and nested conditional statements

C2.4 write programs that include sequential, selection, and repeating events

C2.5 write programs that include the use of Boolean operators, comparison operators, text operators, and arithmetic operators

C2.6 interpret program errors and implement strategies to resolve them

C2.7 write clear internal documentation and use coding standards to improve code readability

C3. Modularity and Modification

demonstrate an understanding of program components and modules

Specific expectations

By the end of this course, students will:

C3.1 analyze existing code to understand the components and outcomes of the code

C3.2 modify an existing program, or components of a program, to enable it to complete a different task

C3.3 write subprograms, and use existing subprograms, to complete program components

C3.4 write programs that make use of external or add-on modules or libraries

C3.5 explain the components of a computational artifact they have created, including considerations for reuse by others

Glossary

Arithmetic operators

Arithmetic operators include: addition, subtraction, multiplication, division, modulo, and exponentiation. These operators are used to represent mathematical expressions in programs.

Boolean operators

Boolean operators are operators such as AND, OR, and NOT. These operators are used in conditional statements to make decisions in programming.

Comparison operators

Comparison operators include: equal to, not equal to, greater than, less than, greater than or equal to, and less than or equal to. These operators are used to compare multiple numeric values. These operators are often used in conditional programming statements.

Computational artifacts

Computational artifacts are objects that a computing system can employ to carry out a task. Programs are the most typical artifact. Others include interactive websites and machine learning models used in artificial intelligence (AI) systems.

Computational thinking

Computational thinking is a problem-solving technique that uses decomposition, pattern recognition, abstraction, and algorithms. Solving a problem using computational thinking concepts and processes often, but not always, involves designing and writing a program.

See also the section "Computational Thinking".

Digital technology

Digital technology is a broad term that refers to electronic systems, devices, and applications that generate, store, process, and/or manage data.

Program

A program is a set of instructions that a computer can execute to provide a desired outcome based on the needs of a user. Programs are usually based on algorithms, and can be written using block-based or text-based programming environments. Programs can also be referred to as applications or software.

Text operators

Text operators include comparison, concatenation, and indexing.