Outline of Course of Study – ICS20

Faculty of Engineering Secondary School
Department of Engineering
Department Head: Pam Lee-Shanok
Teacher: Colleen Maquiling
Course development date: December 12, 2018
Course reviser: Pam Lee-Shanok
Revision Date: January 14, 2019
Course title: Introduction to Computer Studies
Grade: 10
Type: Open
Ministry Course Code: ICS2O
Credit value: 1.0 credit
Ministry curriculum policy documents:

- The Ontario Curriculum, Grades 10 to 12: Computer Studies, 2008 (revised)
- Ontario Schools, Kindergarten to Grade 12: Policy and Program Requirements, 2016
- Growing Success: Assessment, Evaluation, and Reporting in Ontario's Schools, Kindergarten to Grade 12, 2010

Prerequisites and corequisites: none

Course Description

This course introduces students to computer programming. Students will plan and write simple computer programs by applying fundamental programming concepts, and learn to create clear and maintainable internal documentation. They will also learn to manage a computer by studying hardware configurations, software selection, operating system functions, networking, and safe computing practices. Students will also investigate the social impact of computer technologies, and develop an understanding of environmental and ethical issues related to the use of computers.

Overall Curriculum Expectations

By the end of this course, students will:

A. UNDERSTANDING COMPUTERS	
A1	describe the functions of different types of hardware components, and assess the hardware needs of users;
A2	describe the different types of software products, and assess the software needs of users;
A3	use the basic functions of an operating system correctly;
A4	demonstrate an understanding of home computer networking concepts;
A5	explain the importance of software updates and system maintenance to manage the performance and increase the security of a computer.
B. INTRODUCTION TO PROGRAMMING	
B1	describe fundamental programming concepts and constructs;
B2	plan and write simple programs using fundamental programming concepts;
B3	apply basic code maintenance techniques when writing programs.
C. COMPUTERS AND SOCIETY	
C1	describe key aspects of the impact of computers and related technologies on society;
C2	describe computer use policies that promote environmental stewardship and sustainability;
C3	describe legal and ethical issues related to the use of computing devices;

C4	describe postsecondary education and career prospects related to computer studies.

Outline of Course Content

Unit 1: Basic Computer Hardware, Operating and Networking Systems	16 hours
Students will learn about basic computer hardware components and various operating systems. They will also explore computer networks and security by completing exercises on an online network simulation.	
Unit 2: Computers and Society	**12 hours**
Students will learn about the social and environmental impacts of computer use. Students will also explore legal and ethical issues relating to the use of computing devices. In this unit, students will be required to complete a simple research project.	
Unit 3: Introduction to Programming I	**30 hours**
Students will learn how to plan the content of a program by using visual problem-solving models. Students will also learn about basic programming concepts: ● Data types ● Boolean and comparison operators ● Variables ● Selection statements	
Unit 4: Introduction to Programming II	**24 hours**
Students will learn how to use a programmable circuit board to build simple electronic applications. They will also learn how to properly use looping and nested structures to design and program more advanced applications.	
Unit 5: Electronic Application: Final Project	**32 hours**
Students will be introduced to the engineering design process and will learn how to validate computer programs using test cases. They will also learn how to properly document written code to a specific set of standards. In this unit, students are required to design and build an electronic application with a Raspberry Pi microcontroller and present their project.	

Teaching & Learning Strategies

This course is intended to give high school students a good understanding of application development and best practices around it. Our goal is to keep students engaged and provide them with an understanding of industry standards prior to entering the realm of undergraduate education. Students will have the opportunity to take part in class discussions in regards to the technologies that they will manipulate and develop with.

The teacher will demonstrate to students how to setup development environments and will guide students through environment setup. Then students will have the opportunity to take part in student-led assignments and projects. NodeJS will be the language of choice used to guide students through their experiential learning of development methodologies.

The teacher will elaborate on the proper development practices in order to deliver a learning environment consistent with industry practices, enabling students to understand development patterns. The students will be responsible for completing their assignments and homework required for the course. Students will be given the opportunity to direct their own student-led final assignment. This assignment will have students propose a new feature to add to an existing solution that was designed by the teacher. The teacher will either approve or propose an alteration to the feature before students can start working on it. Once the students have determined the feature they want to implement, they will begin working together in groups to develop their feature leveraging the technologies and methodologies learned in class. This project aims specifically to have students develop a set of problem solving skills in regards to development.

It is a common practice in development to give developers access to project management systems, source control, integrated development environments and continuous integration. These are systems that students will have access to in order to develop an application that is using industry practices. With this students will also be using these technologies to help them write and test algorithms.

Students will be continuously assessed through the review of their system tests, their continuous integration environment, and their code. Students will have a major project submission at the end of the course testing their ability to add a single piece of functionality to an already existing system.

Strategies for Assessment & Evaluation of Student Performance

Assessment, evaluation, and reporting of student achievement will be based on the policies and practices outlined in the following Ministry's policy document Growing Success: Assessment, Evaluation, and Reporting in Ontario Schools, 2010.

Students will be evaluated based on the overall expectations of the course through the achievement charts in The Ontario Curriculum, Grades 10 to 12: Computer Studies, 2008 (revised), as outlined in this document

The Ministry of Education's document Growing Success: Assessment, Evaluation, and Reporting in Ontario Schools outlines policies for measuring and communicating achievement. Levels of achievement are defined as follows:

Level	Percentage	Achievement
Level 1	50—59%	Represents achievement that falls much below the provincial standard. The student demonstrates the specified knowledge and skills with limited effectiveness. Students must work at significantly improving learning in specific areas, as necessary, if they are to be successful in the next grade/course
Level 2	60—69%	Represents achievement that approaches the provincial standard. The student demonstrates the specified knowledge and skills with some effectiveness. Students performing at this level need to work on identified learning gaps to ensure future success
Level 3	70—79%	Represents the provincial standard for achievement. The student demonstrates the specified knowledge and skills with considerable effectiveness. Parents of students achieving at level 3 can be confident that their children will be prepared for work in subsequent grades/courses.
Level 4	80—100%	Identifies achievement that surpasses the provincial standard. The student demonstrates the specified knowledge and skills with a high degree of effectiveness. However, achievement at level 4 does not mean that the student has achieved expectations beyond those specified for the grade/course.

Seventy percent (70%) of the evaluation is based on daily classroom work and will be determined through a variety of methods, as outlined in the table below. Thirty percent (30%) of the evaluation will be based on a final design project which includes a prototype and presentation. This final evaluation allows the student the opportunity to demonstrate comprehensive achievement of the overall expectations of the course.

Teachers will use "assessment for learning" and "assessment as learning" practices to help students identify: where they are in relation to the learning goals and what next steps they need to take to achieve the goals.

This ongoing feedback will help prepare students for "assessment of learning", the process of collecting and interpreting evidence for the purpose of summarizing learning at a given point in

time, to make judgments about the quality of student learning on the basis of established criteria, and to assign a value to represent that quality.

Assessment breakdown for ICS2O:

	Percentage of grade	Overall Expectation(s)
Formative Assessment, 70% of final grade		
Identifying Computer Components and Functions	5%	A1
Network Security Exercise	5%	A4, A5
Assignment 1: Flowchart	5%	B2
Assignment 2: Variables	5%	B1
Assignment 3: Strings	5%	B2, B3
Number Guesser Project	10%	B1, B2, B3
Systems Quiz	5%	A1, A2, B1, B2
Computers and Society Presentation	10%	C1, C2, C3, C4
Raspberry Pi Project	10%	B1, B2, B3
Assignment 4: Arrays	5%	B2, B3
Assignment 5: Validation Using Test Cases	5%	B3
Summative Assessment, 30% of final grade		
Final evaluation: Design Project ● Prototype (15%) ● Presentation (10%) ● Documentation (5%)	30%	A1,A5, B1, B2, B3, C1,

Considerations for Program Planning

Instructional Approaches

In computer studies teachers will be using projects as a means for students to gain knowledge and learn new skills such that they can achieve the course expectations. This type of course will give students ample opportunities to collaborate in teams and work cooperatively while working through design and programming problems. When students are actively engaged in experiential learning, then tend to build longer-lasting skills and better retain knowledge.

Through hands-on lessons, students will be given the opportunity to work individually and in teams. Teachers will model new skills, offering direction and support until students are confident in using those skills independently. Teachers will model good program design and good coding practices to set students off on the right foot, and giving them ample time to practice new skills.

By differentiating instruction, teachers will address the various learning preferences and individual needs of their students. He/She will provide examples followed by practice allowing students to learn a variety of concepts, skills, procedures, and processes. With the support of teacher modelling and ample time for practice, students will gain understanding and develop new skill sets in manageable chunks. This scaffolding approach will provide students with the support they need to reach manageable objectives.

Role play and the use of real-world examples will allow students to better grasp the abstract concepts taught in computer studies. Kinesthetic activities can also be incorporated into the classroom such that students better retain new knowledge, such as the concept of binary numbers. Students will have the opportunity to work collaboratively in groups, where each student has a specific role on the team. For instance, in pair programming one student can type out the code while the other student dictates the code and comments, until they switch roles.

Teachers have the freedom to choose the type of programming language for the course (such as object-oriented or procedural) as well as the language itself. It is important for teachers to note that students who recognize the value and relevance of what they are doing will be self-motivated to achieve the course expectations and take an active role in their learning. Selecting appropriate teaching methods and learning activities can have a significant positive effect on students' attitudes toward computer studies.

The Importance of Current Events in Computer Studies

The discussion of current events and emerging technologies stimulates student interest and will be embedded into the computer studies curriculum. It enhances the relevance of the curriculum and helps students connect their in-class lessons with real-world events or situations. Embedding current events into the lessons is an effective instructional strategy for implementing many course expectations into the curriculum.

The Role of ICT in Computer Studies

Information and communications technologies (ICT) tools allow teachers to expand their instructional strategies and support student learning. These tools include Internet websites, word-processing programs, and multimedia resources. These tools help students collect, organize, and present data for reports and presentations. They also enable students to connect with each other and the world to be able to share ideas and collaborate on projects.

Students will be encouraged to use ICT tools for most of the course in order to learn new skills and communicate their learning. Students will be using PowerPoint, for instance, to present their design projects to the class.

With the power of the Internet comes potential risks such as privacy, safety, and abuse of technology in the form of bullying or other malicious acts. Students must be made aware of these issues and teachers will model appropriate behaviour in their instruction. Teachers can also make use of ICT tools in their day-to-day teaching practice of curriculum design and in-class teaching.

Planning Computer Studies Programs for Students with Special Education Needs

Classroom teachers have a duty to ensure that all students in their class have the opportunity to learn and succeed regardless of their special education needs. *Special Education Transformation: The Report of the Co-Chairs with the Recommendations of the Working Table on Special Education, 2006* promotes a set of beliefs that should guide program planning for students with special education. These beliefs include:

- All students can succeed.
- Universal design and differentiated instruction are effective and interconnected means of meeting the learning or productivity needs of any group of students.
- Each student has his or her own unique patterns of learning.
- Classroom teachers need the support of the larger community to create a learning environment that supports students with special education needs.
- Fairness is not sameness.

Teachers are encouraged to develop their program plan in accordance to their students' diversity of strengths and abilities. This can be achieved through a myriad of ways including: assessing each student's prior knowledge and skills, providing ongoing assessment, and allowing for flexible groupings. By assessing each student's current achievement level and weighing that against the course expectations, the teacher can determine if the student will be requiring any combination of: accommodations, modified expectations, or alternative

expectations. If the student requires accommodations, modified expectations, or both, the information must be recorded in their Individual Education Plan (IEP).

Students Requiring Accommodations Only

Accommodations that are required by students must be identified on their IEP. Differentiated instruction and universal design lend themselves well to providing accommodations for students. Students will still be evaluated on the curriculum course expectations and achievement levels communicated by the Ministry.

There are three types of accommodations:

- Instructional accommodations: Teachers change the way in which lessons are taught including integrating technology and using different styles of presentation.
- Environmental accommodations: This includes a change in the learning environment whether it be classroom seating by location or group, or lighting.
- Assessment accommodations: These allow students to demonstrate their learning in a different way. For instance, they may be given the opportunity to give oral answers to written questions or they may be given more time to complete an assignment or test.

Students Requiring Modified Expectations

Modified expectations that are required by students must be identified on their IEP. For the most part, these expectations will be based on the regular course expectations but the number and/or complexity will differ. Modified expectations are specific, realistic, and measurable achievements that the student can demonstrate independently, given assessment accommodations.

It is the principal who will decide whether the achievement of the modified expectations constitutes successful completion of the course and whether the student is eligible to receive a credit for the course; this decision must be communicated to the student and their parents.

When course expectations are not extensively modified and it is expected that the student can achieve most of them, the modified expectations should determine how the required knowledge and skills differ from those identified in the course expectations. In the case, if the student is working toward a credit for the course, the IEP box must be checked on the Provincial Report Card.

With extensive modifications to expectations such that achievement of them is not expected to result in a credit, the expectations should identify the precise requirements or tasks on which the student’s performance will be evaluated and which will be used to determine the student’s mark

on the Provincial Report Card. The IEP box must be checked and the appropriate statement from the *Guide to the Provincial Report Card, Grades 9-12, 1999* (p. 8) must be added. Modified expectations must be reviewed in relation to the student's progress at least once each reporting period, and must be updated as necessary.

Program Considerations for English Language Learners

Schools in Ontario have a very diverse and multicultural student population, such that 20% of students have a language other than English as their first language. These English language learners may be recent immigrants or refugees while others may be born in Canada into a family whose primary home language is either not English or is an English dialect differing significantly from the English taught in Ontario schools. Teachers must be mindful that many of these students are entering a new linguistic and cultural environment at school.

During their first few years in an Ontario school, English language learners pay receive support through English as a Second Language (ESL) programs or English Literacy Development (ELD) programs. ELD programs are primarily for newcomers who arrive with significant gaps in their education, often due to limited opportunities (in terms of education and literacy) in their home country.

It is important that teachers recognize the orientation process whereby English language learners adapt to a new social environment and language. Some may be very quiet at first, using body language rather than speech and/or limited verbal communication to convey their thoughts. These students thrive in a safe, supportive, and welcoming environment. As the students learn to speak English, it is important to note that oral fluency is not a good indicator of the student's literacy development and vocabulary.

It is the shared responsibility of the classroom teacher, the ESL/ELD teacher (where available), and other school staff to help in the development of students' English. Volunteers and peers may also provide significant support. Teachers are required to adapt their instruction to facilitate the success of their English language learner students. These adaptations may include:

- Modifying some or all course expectations such that they are challenging yet achievable given the student's English proficiency
- Using a variety of instruction strategies, such as visual cues, pre-teaching vocabulary, offering peer tutoring
- Using a variety of learning resources, such as bilingual dictionaries, visual material, simplified text
- Modifying assessments, such as giving extra time, offering the choice of demonstrating skills/knowledge orally or in writing, assigning cloze sentences instead of essays

When learning expectations are modified for an English language learner, it must be clearly indicated on their report card.

Antidiscrimination Education in Computer Studies

The Faculty of Engineering Secondary School abides by the University of Ottawa's Violence Prevention Policy and Prevention of Harassment and Discrimination Policy. These policies encourage staff and students to show respect for diversity in the school and the wider society. The policies aim to provide a safe learning environment, free from violence, harassment, and discrimination.

Differentiated instruction will be at the core of curriculum planning. By assessing each individual student's abilities, background, interests and learning styles, teachers can design their lessons based on the needs of their diverse students. The course content (what is being taught), process (how it is taught), and product (how students demonstrate their learning) will be designed in relation to the students' needs.

Generally in technical courses such as computer science there is a clear gender disparity. Studies have shown that female students are often drawn to courses that have a societal aspect to them, rather than just abstract learning. It may be helpful for teachers to offer projects and activities that have a clear and meaningful societal application. For instance, instead of being asked to design a robotic arm (whose purpose is unknown), teacher can give students the option of designing an assistive device. Differentiated instruction offers students a choice from a range of activities or allows them to select their own projects; by giving students the power to choose their own topic, they can select something that most interests them and become more invested in the project.

In 2019 we will have a wonderful teacher (doubling as a role model) for our classes. Both our Introduction to Computer Studies (ICS2O) and Introduction to Computer Science (ICS3U) classes will be taught by a female visible minority. She will be bringing the students on a field trip to a tech company where they will be exposed to a myriad of jobs in the tech industry and will see a diverse workforce in the field.

Environmental Education and Computer Studies

It is important for students to understand their environmental impact in the world and how they can better the environment they are living in. It is the duty of the teacher to integrate environmental education into their curriculum planning such that students understand their personal responsibility to the environment and their role in society.

Environmental education can be integrated into the classroom in a variety of ways. In selecting their projects, students can go the environmental route and select a project that is directly linked to environmental impact, such as a simulation of a healthy ecosystem or the consequences of an oil spill. Additionally, students can focus on the environmental impact of computer use by learning about the safe handling and disposal of materials used in the manufacturing of computer components. By implementing strategies to reduce, reuse and recycle, students can learn about government agencies and community partners that support such practices. This will give students the opportunity to develop critical thinking skills and responsible practice with respect to environmental implications of their selected project.

Programming projects can be used to address environmental-focused course expectations. For instance, students can program a survey that assesses people's environmental awareness as it relates to the use of computers. The program could calculate the awareness and suggest strategies or provide feedback to users.

Literacy, Mathematical Literacy, and Inquiry/Research Skills

Many activities in the computer studies curriculum requires students to practice and develop oral, written, and visual literacy skills. Students will be required to brainstorm ideas and effectively communicate them to their team members. They will need to be able to justify their choices for decisions taken in the design process and will need to be able to communicate them clearly to their audience in an oral presentation with visual support. They will be required to compose written reports on their progress and outline the steps taken during the design process in order to effectively convey their message to the reader. Students will be learning specialized terminology which they will be expected to use appropriately and precisely in their communication.

In developing programs, students will build on their mathematical literacy. Students will be required to communicate clearly and concisely through the use of tables, diagrams, and/or flow charts. Many components of the computer studies curriculum emphasize students' ability to interpret and use symbols and charts.

In conducting research for their projects, students will be required to explore a variety of possible solutions to their challenge, analysing the context of their data and properly interpreting it. They will be required to analyse the source of their information, determine its validity and relevance, and use it in appropriate ways. Teachers can support students by guiding them toward reputable sources including peer-reviewed journals. The ability to locate, question, and evaluate information allows a student to become an independent, lifelong learner.

The Ontario Skills Passport and Essential Skills

The Ontario Skills Passport (OSP) is a web-based service that can track students' Essential Skills (such as reading, writing, and problem solving) and work habits (such as working safely and being reliable). These skills and work habits are easily transferable from school to work and are useful for employers looking to assess potential candidates for cooperative education placements. The OSP is also useful for students looking to assess, build, document, and track their skills through their educational, professional, and personal experiences. More information about the OSP can be found on the ministry website, http://skills.edu.gov.on.ca.

Career Education

In this era of technological innovation with rapidly evolving technologies, employers are always on the lookout for candidates with strong technical skills who can problem-solve effectively, think critically, and work collaboratively. These are the exact skills that will be developed through computer studies courses. In going through the design process, students will develop skills in: research, analysis, creativity, problem-solving, design, and presenting. They will practice these skills through both independent and group work.

Cooperative Education and Other Forms of Experiential Learning

Cooperative education and other forms of experiential learning, such as job shadowing, work experience, and field trips, allow students to apply the skills they've learned in the classroom to real-word work environments. They help students learn about the possible careers and employment opportunities in various fields of work, as well as broadening their knowledge of workplace practices and employer-employee relationships.

Students who choose a computer studies course as the related course for two cooperative education credits are able, through this packaged program, to meet the group 1, 2, and 3 compulsory credit requirements for the OSSD.

Teachers must assess the health and safety of placements and ensure that their students understand their rights as they relate to health and safety, privacy and confidentiality, and abuse and harassment in the workplace.

All cooperative education and other workplace experiences will be provided in accordance with the ministry's policy document *Cooperative Education and Other Forms of Experiential Learning: Policies and Procedures for Ontario Secondary Schools, 2000*.

Planning Program Pathways and Programs Leading to a Specialist High Skills Major

Computer studies courses are well suited for programs leading toward a Specialist High Skills Major (SHSM) or programs leading toward an apprenticeship or workplace destination. Computer studies courses can also be combined with cooperative education credits in order to provide the workplace experience necessary for some SHSM programs, apprenticeships, and workplace destinations. SHSM programs would also include sector-specific learning opportunities offered by employers, skills-training centres, colleges, and community organizations.

Health and Safety in Computer Studies

The most common health and safety concerns associated with repeated computer use are eye strain and musculoskeletal injuries (including repetitive strain injuries). Teachers will ensure that work stations are ergonomic and that students maintain good posture and take frequent eye and body breaks. Students will also be taught about emotional and health risks common among heavy computer users, particularly social isolation.

Teachers will assess any risks associated with field trips including the transportation risks and risks at the visiting location and communicate these risks with parents and students. When activities take place outside of the predictable classroom environment, it is the teacher's duty to ensure the health and safety of students is maintained.

Resources

No textbook is required for this course, although the teacher will supply articles and blogs for students to read in order to extend their knowledge of the course. Students will be given access to all course material in class and will be given access to computer laboratories during and after class hours in order to continue their learning. Students will also be given access to any system required for the course, such as Travis CI, GitHub, Eclipse and SourceTree.