Stouffville District Secondary School

ICS4U-Course Outline

Board:	YRDSB
School:	SDSS
Curriculum Leader:	Mr. A. Krnic
Developing Teacher:	Mr. A. Krnic
Date of Revision:	**Sept. 2018**
Course Title:	Computer Science, Grade 12, University
Grade:	12
Course Code:	ICS4U
Credit Value:	1.0
Pre-requisite	None
Textbook:	**None**
Resources:	**Eclipse, JAVA**
Supplementary Resources:	The Ontario Curriculum, Grades 10 to 12: Computer Studies, 2008 (revised) handouts, computer files, exemplars, PowerPoint presentations
Course Description:	This course introduces students to computer science. Students will design software independently and as part of a team, using industry-standard programming tools and applying the software development life-cycle model. They will also write and use subprograms within computer programs. Students will develop creative solutions for various types of problems as their understanding of the computing environment grows. They will also explore environmental and ergonomic issues, emerging research in computer science, and global career trends in computer-related fields.

Learning Skills

Assessment of the learning skills will be done on an ongoing basis throughout the academic year by
observations of students at work, checklists and interviews. This will include:

Classwork/homework (Work habits, homework and organization)
Completed work and seeking assistance (Organization and initiative)
Persistence and independence at tasks (Working independently and initiative)
Extension of task (Organization and initiative**)**
Achievement of group goals (Team work)

Assessment Strategies

A variety of teaching/assessment strategies to address students' needs will be used during the school year.
Formative assessments will be ongoing through out the academic year. These *may* include:
- Diagnostic assessment
- Formative assessment

 Performance assessment
 Portfolio assessment
 Rubrics
 Checklists

Term Summative Evaluations (70% Term Work)

 Tests, quizzes, projects, assignments and other forms of term summative evaluations will occur throughout the academic year and at the end of units of work as outlined in the accompanying course
outline.
 Students will be provided with reasonable opportunities to master skills relating to the achievement of
the curriculum expectations before assessment and evaluation occurs.
 Major evaluations will be announced at least one week in advance.
 Accommodations will be made for school activities, statutory holidays, religious days, cultural days,
sports events and other occurrences that may impact on any scheduled evaluation. It is the student's
responsibility to notify teachers of such absences in advance and to make up missed work.
 Absence on the day of an evaluation must be documented. If a student must miss an evaluation, s/he is
expected to:
a. notify the teacher before the absence to arrange for an alternative date to make up the evaluation; or
b. in case of illness or unexpected absence, present a note to the teacher, signed by a parent or guardian, immediately upon their return to explain the absence. An alternate evaluation will then be scheduled at a mutually convenient time
 The Late Policy applies to all assignments and evaluations and will be discussed in class.
 Cheating will not be tolerated and will be dealt with appropriately.

Final Mark Calculation

Final marks will be calculated as follows:

Term Work: 70% *Levels of Achievement:*

Knowledge and Understanding: 25%	Level 1:50 - 59%
Application: 25%	Level 2:60 – 69%
Thinking: 10%	Level 3:70 – 79%
Communication: 10%	Level 4: 80 - 100%

Final Summative Evaluations: 30%

Final Programming Project 15%
Examination 15%

Reporting

Report #1 100% Term Work
Report #2 100% Term Work
Final Report 70% Term Work + 15% Final Project + 15% Exam (Cumulative)

Unit Overviews

Unit 1: Programming Concepts
Periods: 20

Unit Description
This unit focuses on basic computer and information science skills. Students identify hardware components, research ergonomic considerations, practise file management skills, access resources through local and wide area networks, and research the evolution of programming languages. They develop skills for success in the computer and information science environment.

This unit also focuses on the basic and advanced programming structures learned in the grade 11 computer science course. Students write simple programs, using variable assignment, repetition and decision structures, arrays, functions, algorithms and develop effective testing, validating, and documenting skills. They also explore roles of effective communicators and reflective thinkers when following a problem-solving model (e.g., user inputs a series of marks, each value is validated, the average is calculated, and a grade is assigned).

Unit 2: Topics in Computer Science
Periods: 10

Unit Description
This unit includes an investigation into societal issues involving computer technology. It involves an exploration of careers in computer studies. Students also examine issues surrounding privacy, security, and ethical use of information.

Unit 3: Designing and Analysing Algorithms
Periods: 15

Unit Description
This unit focuses on using problem solving strategies in the computer science field with an emphasis on exploring and implementing common computer science algorithms.

Unit 4: Modular Programming: OOP (Learning the Concepts behind Object Oriented Programming)
Periods: 15

Unit Description
This unit focuses on the advanced OOP features of programming. Students learn about the key principles in OOP like inheritance, polymorphism, encapsulation and immutable objects. Students are expected to incorporate as many of these complex programming features in their final summative project.

Unit 5: Software Development
Periods: 20
Students will spend the last few weeks of class designing and developing an application that incorporates most, if not all, the concepts taught in the course. Example ideas include the following: video games, multimedia organizers, educational quizzes etc. This project comprises 15% of the course mark.

Course Requirements:
Students must bring to a flash drive. Flash drives should be labelled with the student's name, class and teacher's name. A simple code editor and FlashDevelop are the primary software used in the class. The vast majority of projects will require the use of these programs. You may download and use these for free. See your teacher's website for links.