Senior 2 (20S), Senior 3 (30S), and Senior 4 (40S) Computer Science

Manitoba Curriculum Framework of Outcomes

Manitoba
Building for the Future

SENIOR 2 (20S), SENIOR 3 (30S), AND SENIOR 4 (40S) COMPUTER SCIENCE

Manitoba Curriculum Framework of Outcomes

2004

Manitoba Education, Citizenship and Youth

Manitoba Education, Citizenship and Youth Cataloguing in Publication Data

004.0712 Senior 2 (20S), Senior 3 (30S), and Senior 4 (40S)
computer science : Manitoba curriculum
framework of outcomes

Includes bibliographical references.
ISBN 0-7711-3272-7

1. Computer science—Curricula. 2. Computer
science—Study and teaching (Secondary)—Manitoba.
3. Electronic data processing—Curricula. 4. Electronic
data processing—Study and teaching (Secondary)—
Manitoba. I. Manitoba. Manitoba Education, Citizenship
and Youth.

Every effort has been made to acknowledge original sources and to comply with copyright law. If cases are identified where this has not been done, please notify Manitoba Education, Citizenship and Youth. Errors or omissions will be corrected in a future edition.

This document is available on the Manitoba Education, Citizenship and Youth website at <http://www.edu.gov.mb.ca/ks4/cur/cs/index.html>.

Acknowledgements

Manitoba Education, Citizenship and Youth gratefully acknowledges the contributions of the following individuals in the development of *Senior 2 (20S), Senior 3 (30S), and Senior 4 (40S) Computer Science: Manitoba Curriculum Framework of Outcomes*.

Development Team Members

Kathy Athayde	Kildonan-East Collegiate	River East-Transcona School Division
Normand Châtel	Collège Béliveau	Louis Riel School Division
Bob Dagg	Elton Collegiate	Rolling River School Division
Sylvain Delaquis	École Gabrielle-Roy	Division scolaire franco-manitobaine
Dino Difabrizio	Technical Vocational High School	Winnipeg School Division
Cheryl Klimack	Crocus Plains Regional Secondary School	Brandon School Division
Kevin Rowan	Fort Richmond Collegiate	Pembina Trails School Division

Industry Representatives

Geoff Bresch	Manitoba Information Technology Industries Inc.
Scott Greenlay	Greenridge Business Systems Corporation
Norm Lee	MindSet, Manitoba Network for Science and Technology
Wendy Magahay	Applications of Working and Learning (AWAL) National Project, Camosun College
Bruce Popham	Manitoba Solution Centre, EDS Canada Inc.
David Tetlock	Investors Group

AWAL Workshop

During the development of this document, development team members participated in an Applications of Working and Learning (AWAL) workshop facilitated by Wendy Magahay. Team members visited worksites and interviewed employees about the essential skills, knowledge, and attitudes required in careers in computer programming and project management. The following organizations volunteered to participate and provided access to staff for this workshop.

EDS Canada

Encore Business Solutions

The Great-West Life Assurance Company

Greenridge Business Systems Corporation

Manitoba Hydro

Workers Compensation Board of Manitoba

Manitoba Education, Citizenship and Youth Staff
School Programs Division

Lee-Ila Bothe	Coordinator	Production Support Unit Program Development Branch
Diane Cooley	Project Manager	Curriculum Unit Program Development Branch
Darryl Gervais	Project Leader/Writer	Distance Learning and Information Technologies Unit Program Development Branch
Michelle Larose-Kuzenko	Consultant	Bureau de l'éducation française Division
Susan Letkemann	Publications Editor	Production Support Unit Program Development Branch
Sam Steindel	Coordinator	Distance Learning and Information Technologies Unit Program Development Branch
Lindsay Walker	Desktop Publisher	Production Support Unit Program Development Branch

Contents

Introduction

"The use of information technology will help enable all students to solve problems, improve their personal performance, and gain the critical and abstract thinking skills necessary to become lifelong learners and contributing members of their communities" (Manitoba Education and Training, *Technology As a Foundation Skill Area,* 7).

Computer Science Framework

This document provides a framework of learning outcomes for students completing Senior Years computer science courses. The framework serves as a basis for the development of Senior 2 (20S), Senior 3 (30S), and Senior 4 (40S) Computer Science courses. It replaces the former curriculum documents: *Computer Science 205* (1983) and *Computer Science 305* (1984).

The emphasis in computer science courses is on students learning to solve problems, accomplish tasks, and express creativity, both individually and collaboratively. Students will learn programming techniques and the syntax of one or more programming languages. More importantly, students will learn to adapt to changes in programming languages and learn new languages as they are developed.

Information and Communication Technologies

Effective integration of information and communication technologies into all curricula assists students in developing the abilities to use, manage, and understand the technologies. Prior to the start of computer science courses, students should have developed the competencies identified for the **skill development stage** of the "Information Technology Literacy Continuum" outlined in *Technology As a Foundation Skill Area* (hereafter referred to as TFS—see Appendix A, pages 17 to 24). Students studying computer science are at the **application and extension stage** of the continuum outlined in TFS.

Computer science courses provide students with opportunities to apply previous learning and extend their information and communication technology skills. Specifically, "students should have opportunities to explore various programming languages and computer-controlled devices and systems through specialized Senior Years courses" (TFS, 23). Computer science courses enable students to explore and develop skills in solving problems and prepare them for further studies at college or university.

Access to Computer with Programming Language

Many of the computer science learning outcomes can be achieved only by programming a computer. All students must, therefore, have access to a computer with a modern programming language installed. The computer science framework does not prescribe a specific language. The teacher and students will decide which modern programming languages students will learn.

Learning, Teaching, and Assessing

The Student As Learner

Students are curious, active learners who have individual interests, abilities, and needs. It is crucial to recognize how students learn and how learning can be assessed. Learning involves the process of linking newly constructed understanding with prior knowledge and adding new contexts and experiences to current understanding. The learning outcomes in computer science are designed to build upon previous learning. Each level of computer science builds upon the previous level.

Students must become independent learners if they are to pursue further studies and careers in computer science. Programming languages and techniques continue to evolve rapidly. Students are expected to learn new information and adapt to changes continually. To ensure that students become lifelong learners, it is imperative that they become increasingly engaged in planning, developing, and assessing their own learning experiences. They must have opportunities to work with other students, to initiate investigations, to communicate their findings, and to complete projects that demonstrate their learning.

At the beginning of a block of instruction, teachers, together with students, need to identify expected learning outcomes and establish performance criteria that correspond with provincial learning outcomes. This communication between teachers and students assists in planning for instruction, assessment, evaluation, and reporting. It also helps identify clearly what students need to accomplish, and thus facilitates the learning process.

When students are aware of expected learning outcomes, they will be more focused on the learning and more likely to assess their own progress and achievement. Furthermore, they can participate in creating appropriate assessment and evaluation criteria. Assessment methods must be valid, reliable, and fair to students.

The Teacher As Facilitator

While teachers are experts in facilitating groups and using instructional and assessment strategies that help students learn, they need not be experts in programming a computer. If computer science courses are to remain current and relevant, however, teachers must be open to and accepting of innovations in computer science and willing to learn with students.

As facilitators, teachers

- assist students as they navigate through information made available by information and communication technologies and other sources
- direct students as they gather, organize, analyze, and present their findings
- help students recognize, develop, focus, refine, consolidate, and extend their knowledge, skills, and competencies

Prerequisites

No prerequisites are specified for computer science courses. The framework is designed so that the computer science courses build upon the outcomes from the previous courses. Schools may choose to offer one, two, or three full credits in computer science. If the school does not offer the course at Senior 2, then learning outcomes from 20S will need to be incorporated into the 30S course. Similarly, if the school does not offer the course at Senior 3, then outcomes from 30S will need to be incorporated into the 40S course.

Reporting

All computer science courses (20S, 30S, and 40S) are full-credit courses. As indicated in the Department's *Subject Table Handbook*, the computer science course code is 0280. Use this number when completing Professional School Personnel (PSP) forms and entering student records into the Student Records System (SRS).

Notes

Senior 2, Senior 3, and Senior 4
Computer Science Learning Outcomes

Computer Science Learning Outcomes

This framework contains the general and specific learning outcomes for computer science courses. The learning outcomes are concise statements of the learning that students are expected to demonstrate in computer science courses by the end of Senior 2, Senior 3, and Senior 4. This learning includes:

- **Knowledge:** Students need to know facts, concepts, principles, and generalizations. The knowledge learned in computer science includes the vocabulary and function of computer hardware, the vocabulary and syntax of computer-programming languages, programming standards and conventions, project-management practices, health and safety issues, and career information.
- **Skills and Strategies:** Students need to know and apply processes and strategies in developing skills. The skills include problem solving, critical thinking, metacognition, communication, and teamwork.
- **Attitudes:** Students need to develop attitudes and habits that include setting goals, thinking strategically in approaching a task, considering personal health and safety, acting ethically and morally, and reflecting on their own performance.

General Learning Outcomes

The general learning outcomes (GLOs) are broad statements describing student learning. The general outcomes are interrelated and interdependent. Each outcome is to be achieved through a variety of learning strategies and experiences.

The four general learning outcomes of the computer science framework reflect the four foundation skill areas:

1. **Human Relations:** Students will demonstrate tolerance, teamwork, leadership, and responsible, ethical, and moral behaviour.
2. **Literacy and Communication:** Students will demonstrate effective communication skills in listening, speaking, reading, writing, viewing, and representing.
3. **Problem Solving:** Students will demonstrate appropriate problem-solving skills while seeking solutions to technological challenges.
4. **Technology:** Students will develop the abilities to use, manage, and understand information and communication technologies by exploring programming languages and computer-controlled devices.

Specific Learning Outcomes

Each general learning outcome is elaborated through a sequence of specific learning outcomes, categorized under headings. The specific outcomes (SLOs) are relevant for all students in a variety of learning environments and are cumulative across grades. Students are expected to demonstrate the specific learning outcomes for their current grade while building on and maintaining the outcomes of previous grades.

Guide to Reading the Learning Outcomes

The learning outcomes for each computer science course (20S, 30S, and 40S) are identified in separate columns of a table presented on the following pages. Read each page vertically for learning outcomes expected at the end of each course. Read each page horizontally for developmental flow of learning outcomes from course to course. An arrow (➠) indicates that a learning outcome from one course is to be reviewed, reinforced, and maintained in the subsequent course(s).

Each learning outcome is numbered:

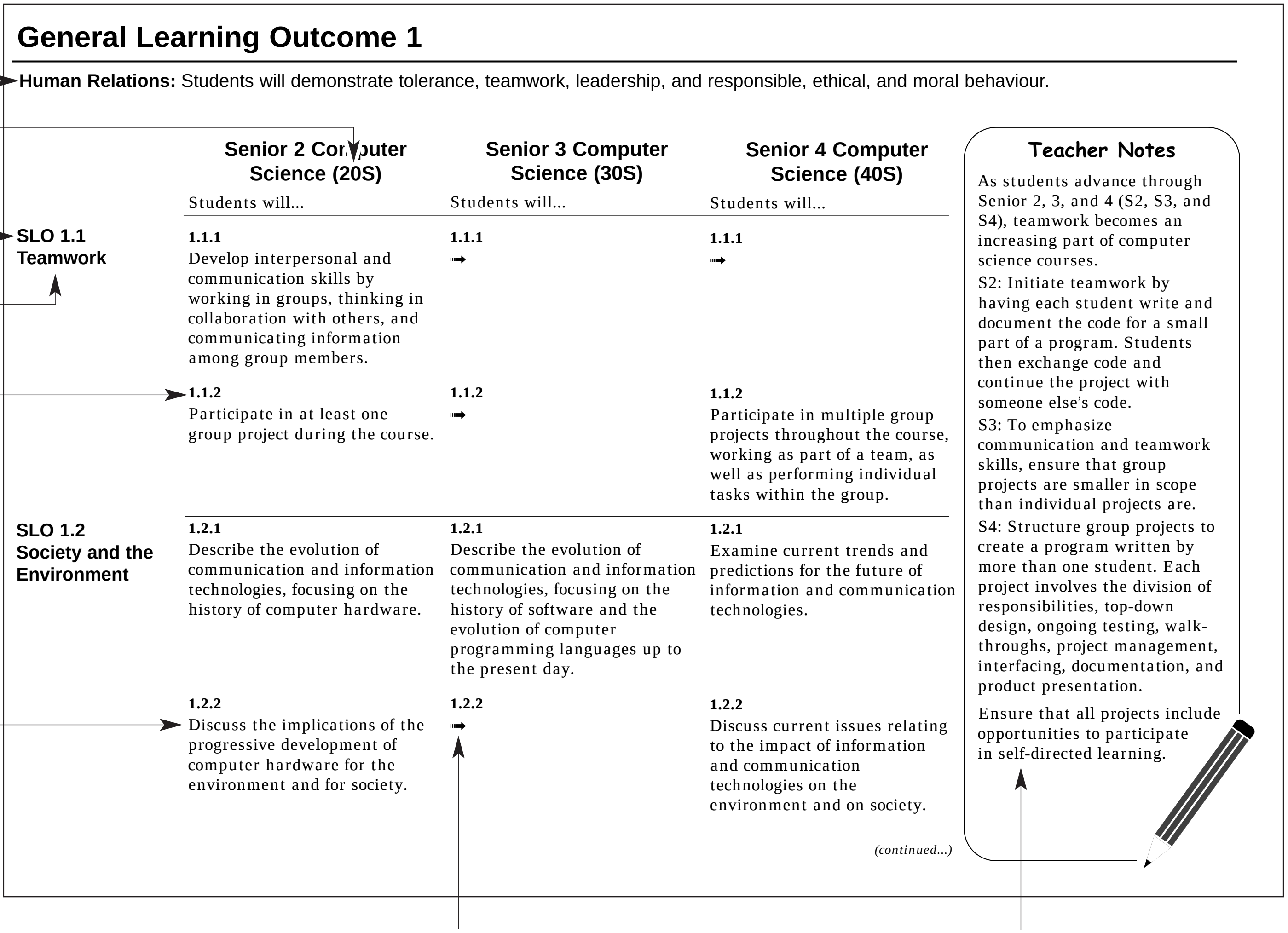
- The first digit indicates the GLO.
- The second digit indicates the heading for a cluster or group of SLOs.
- The third digit indicates the SLO.

The teacher notes are neither prescriptive nor exhaustive; rather, they are intended to clarify the intent of the learning outcomes and suggest how the outcomes might be achieved in general terms.

Guide to Reading the Codes

Read each page horizontally for developmental flow of learning outcomes from course to course.

Read each page vertically for learning outcomes expected at the end of each course.



General Learning Outcome 1

Human Relations: Students will demonstrate tolerance, teamwork, leadership, and responsible, ethical, and moral behaviour.

	Senior 2 Computer Science (20S)	Senior 3 Computer Science (30S)	Senior 4 Computer Science (40S)
	Students will...	Students will...	Students will...
SLO 1.1 **Teamwork**	**1.1.1** Develop interpersonal and communication skills by working in groups, thinking in collaboration with others, and communicating information among group members.	**1.1.1** ➠	**1.1.1** ➠
	1.1.2 Participate in at least one group project during the course.	**1.1.2** ➠	**1.1.2** Participate in multiple group projects throughout the course, working as part of a team, as well as performing individual tasks within the group.
SLO 1.2 **Society and the Environment**	**1.2.1** Describe the evolution of communication and information technologies, focusing on the history of computer hardware.	**1.2.1** Describe the evolution of communication and information technologies, focusing on the history of software and the evolution of computer programming languages up to the present day.	**1.2.1** Examine current trends and predictions for the future of information and communication technologies.
	1.2.2 Discuss the implications of the progressive development of computer hardware for the environment and for society.	**1.2.2** ➠	**1.2.2** Discuss current issues relating to the impact of information and communication technologies on the environment and on society.

(continued...)

Teacher Notes

As students advance through Senior 2, 3, and 4 (S2, S3, and S4), teamwork becomes an increasing part of computer science courses.

S2: Initiate teamwork by having each student write and document the code for a small part of a program. Students then exchange code and continue the project with someone else's code.

S3: To emphasize communication and teamwork skills, ensure that group projects are smaller in scope than individual projects are.

S4: Structure group projects to create a program written by more than one student. Each project involves the division of responsibilities, top-down design, ongoing testing, walk-throughs, project management, interfacing, documentation, and product presentation.

Ensure that all projects include opportunities to participate in self-directed learning.

GLO—
General Learning Outcome

Course Designation

SLO—
Specific Learning Outcome

Heading for cluster of SLOs.

The **SLO number** applies to each course.

- First digit indicates the GLO.
- Second digit indicates the heading.
- Third digit indicates the SLO.

The **SLO** is expected to be achieved by the end of a given course.

An **arrow** (➠) indicates that an SLO from the previous course is to be reviewed, reinforced, and maintained in the subsequent course(s).

The **teacher notes** clarify the intent of the SLO and suggest how it might be achieved in general terms.

General Learning Outcome 1

Human Relations: Students will demonstrate tolerance, teamwork, leadership, and responsible, ethical, and moral behaviour.

	Senior 2 Computer Science (20S) Students will...	Senior 3 Computer Science (30S) Students will...	Senior 4 Computer Science (40S) Students will...
SLO 1.1 Teamwork	**1.1.1** Develop interpersonal and communication skills by working in groups, thinking in collaboration with others, and communicating information among group members.	**1.1.1** ➠	**1.1.1** ➠
	1.1.2 Participate in at least one group project during the course.	**1.1.2** ➠	**1.1.2** Participate in multiple group projects throughout the course, working as part of a team, as well as performing individual tasks within the group.
SLO 1.2 Society and the Environment	**1.2.1** Describe the evolution of communication and information technologies, focusing on the history of computer hardware.	**1.2.1** Describe the evolution of communication and information technologies, focusing on the history of software and the evolution of computer programming languages up to the present day.	**1.2.1** Examine current trends and predictions for the future of information and communication technologies.
	1.2.2 Discuss the implications of the progressive development of computer hardware for the environment and for society.	**1.2.2** ➠	**1.2.2** Discuss current issues relating to the impact of information and communication technologies on the environment and on society.

Teacher Notes

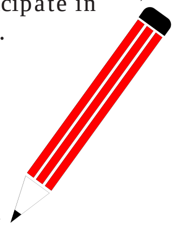
As students advance through Senior 2, 3, and 4 (S2, S3, and S4), teamwork becomes an increasing part of computer science courses.

S2: Initiate teamwork by having each student write and document the code for a small part of a program. Students then exchange code and continue the project with someone else's code.

S3: To emphasize communication and teamwork skills, ensure that group projects are smaller in scope than individual projects are.

S4: Structure group projects to create a program written by more than one student. Each project involves the division of responsibilities, top-down design, ongoing testing, walk-throughs, project management, interfacing, documentation, and product presentation.

Ensure that all projects include opportunities to participate in self-directed learning.



(continued...)

General Learning Outcome 1 (continued)

Human Relations: Students will demonstrate tolerance, teamwork, leadership, and responsible, ethical, and moral behaviour.

	Senior 2 Computer Science (20S)	Senior 3 Computer Science (30S)	Senior 4 Computer Science (40S)
	Students will...	Students will...	Students will...
SLO 1.3 Ethical Behaviour	**1.3.1** Discuss the effects of computer crime, hacking, plagiarizing code, software piracy, virus distribution, and willful destruction of data.	**1.3.1** ➠	**1.3.1** ➠
	1.3.2 Demonstrate responsible security practices while using computers and networks.	**1.3.2** ➠	**1.3.2** ➠

Teacher Notes

Computer science courses teach much more than writing program code. Programs are created to meet a human need.

Industry representatives repeatedly state that human relations are the most important aspect of successful computer programming. Ensure that human relations are developed and reinforced throughout all computer science courses.

General Learning Outcome 2

Literacy and Communication: Students will demonstrate effective communication skills in listening, speaking, reading, writing, viewing, and representing.

	Senior 2 Computer Science (20S) Students will...	Senior 3 Computer Science (30S) Students will...	Senior 4 Computer Science (40S) Students will...
SLO 2.1 Documentation	**2.1.1** Explain the need for and purpose of documentation.	**2.1.1** ➠	**2.1.1** ➠
	2.1.2 Document the program code for all assignments.	**2.1.2** ➠	**2.1.2** ➠
	2.1.3 Assign meaningful names to variables using accepted conventions.	**2.1.3** ➠	**2.1.3** ➠
SLO 2.2 Oral Presentation	**2.2.1** Present an informal explanation of ideas and code to the teacher and other students.	**2.2.1** Plan and deliver a formal individual oral presentation on the solution to a problem.	**2.2.1** Plan and participate in a formal oral group presentation on the solution to a problem.
SLO 2.3 Careers	**2.3.1** Explore careers in information and communication technologies that relate to computer science.	**2.3.1** Research a career in information and communication technologies that relates to computer science.	**2.3.1** Research a career in information and communication technologies that relates to computer science and project management. **2.3.2** Research post-secondary education options for further studies in computer science.

Teacher Notes

Documentation communicates the programmer's intent and supports the use of the product.

S2: Stress the need for adequate documentation from the start, focusing on an explanation of the workings of the program code.

S3: Documentation for all assignments should include a description of the pre- and post-condition of the data for each module within the program.

S4: Documentation expands to include separate written communication that describes the problem that the program solves and instructs the user in the use of the program (for example, organization chart, user manual, help file, and frequently asked questions).

Other students, community members, and Elders can benefit from this research if students share their knowledge through presentations or websites.

Research can include careers and opportunities that are viable in small and/or isolated communities.

(continued...)

General Learning Outcome 2 (continued)

Literacy and Communication: Students will demonstrate effective communication skills in listening, speaking, reading, writing, viewing, and representing.

	Senior 2 Computer Science (20S) Students will...	**Senior 3 Computer Science (30S)** Students will...	**Senior 4 Computer Science (40S)** Students will...
SLO 2.4 Project Management			**2.4.1** Define the term *project*. **2.4.2** Describe the roles of a project manager and project leader and differentiate between the two roles. **2.4.3** Identify the primary reasons why projects succeed or fail. **2.4.4** Prepare a project plan, including a milestone schedule, an action plan, an estimate of resource requirements, a project budget, and a risk analysis. **2.4.5** Resolve project problems. **2.4.6** Motivate team members. **2.4.7** Report project status. **2.4.8** Evaluate project results.

Teacher Notes

Computer programs are rarely created by one person working in isolation. Large computer programs are created by teams of people.

Project management is a large part of a computer programmer's role. Successful project management will ensure that each project meets specifications and is completed on time and within budget.

The project-management skills that students learn in a computer science course are transferable to other courses and situations in their lives.

General Learning Outcome 3

Problem Solving: Students will demonstrate appropriate problem-solving skills while seeking solutions to technological challenges.

	Senior 2 Computer Science (20S) Students will...	Senior 3 Computer Science (30S) Students will...	Senior 4 Computer Science (40S) Students will...
SLO 3.1 Learning to Learn	**3.1.1** Develop skills and strategies for independent learning.	**3.1.1** ➠	**3.1.1** ➠
			3.1.2 Research information on a programming language that is new to them, and learn the syntax independently by creating a new program or recreating an existing program written in a different language.
SLO 3.2 Reasoning and Logic	**3.2.1** Use a structured model for solving problems.	**3.2.1** ➠	**3.2.1** ➠
	3.2.2 Define the terms *algorithm*, *pseudocode*, and *stepwise refinement*.	**3.2.2** Use stepwise refinement.	**3.2.2** ➠

(continued...)

Teacher Notes

Provide students with opportunities and experiences that motivate them and foster a sense of purpose. Encouraging curiosity, self-confidence, and a positive self-concept will help students become capable, self-reliant, and self-motivated people who value learning.

Students require access to learning resources in the school or community and opportunities to learn from peers or mentors.

Model problem solving throughout S2 to S4, emphasizing the importance of analysis and planning. Recognize that it is more natural for some people to approach a problem as a "whole picture" than in a "sequential" manner. Provide less detail and support as students advance through S3 and S4.

S2: Provide the problem and a planned solution. Have students carry out the plan. *(continued...)*

General Learning Outcome 3 (continued)

Problem Solving: Students will demonstrate appropriate problem-solving skills while seeking solutions to technological challenges.

	Senior 2 Computer Science (20S) Students will...	Senior 3 Computer Science (30S) Students will...	Senior 4 Computer Science (40S) Students will...
SLO 3.2 Reasoning and Logic	**3.2.3** Identify the logical series of steps involved in solving a problem.	**3.2.3** ➡	**3.2.3** Decide upon the most appropriate method to solve a problem.
	3.2.4 Confirm an algorithm's logic by hand-tracing a computer program.	**3.2.4** ➡	**3.2.4** ➡
	3.2.5 Use Boolean logic to draw analogies between computer programs and everyday tasks.	**3.2.5** ➡	**3.2.5** ➡
	3.2.6 Graphically represent the solution to non-numeric and numeric problems using – diagrams – pseudocode	**3.2.6** ➡	**3.2.6** ➡
			3.2.7 Describe and recognize the difference between top-down and bottom-up design.

Teacher Notes

S3: Provide the problem. Have students analyze the problem, create a plan to solve it, and design and carry out the solution.

S4: Have students identify the problems and solve them through a team effort, with roles defined and tasks divided among team members. By the end of S4, students will be able to complete all steps of the problem-solving model on their own and as part of a group.

Students may have used a five-step approach, such as the following, to solve problems in mathematics:

1. Understand the problem.
   – analyze the problem
2. Make a plan.
   – organize the steps to solve the problem
   – outline a solution
   – create a prototype
   – create an algorithm
3. Carry out the plan.
   – code the solution
4. Look back.
   – test the code
5. Communicate the solution.
   – present the product

General Learning Outcome 4

Technology: Students will develop the abilities to use, manage, and understand information and communication technologies by exploring programming languages and computer-controlled devices.

	Senior 2 Computer Science (20S) Students will...	**Senior 3 Computer Science (30S)** Students will...	**Senior 4 Computer Science (40S)** Students will...	**Teacher Notes**
SLO 4.1 Programming Standards and Conventions	**4.1.1** Set up a directory and file-management system for storage, retrieval, and portability.	**4.1.1** ➠	**4.1.1** ➠	Model file management by creating a classroom standard for organizing and naming files.
	4.1.2 Explain the need for programming standards and conventions.	**4.1.2** ➠	**4.1.2** Research current programming standards and conventions.	
	4.1.3 Use accepted programming standards and conventions for – documentation – style and readability – naming of variables, subprograms, etc.	**4.1.3** ➠	**4.1.3** ➠	
			4.1.4 Develop and document team standards as part of a group project.	

(continued...)

General Learning Outcome 4 (continued)

Technology: Students will develop the abilities to use, manage, and understand information and communication technologies by exploring programming languages and computer-controlled devices.

	Senior 2 Computer Science (20S)	Senior 3 Computer Science (30S)	Senior 4 Computer Science (40S)
	Students will...	Students will...	Students will...
SLO 4.2 Programming: Data Structures	**4.2.1** Recognize that a particular data structure permits certain operations to be carried out efficiently.	**4.2.1** Recognize how a one-dimensional array makes the solution to some problems possible and makes other problems easier to solve.	**4.2.1** Use a two-dimensional array.
	4.2.2 Identify and use appropriate data types.	**4.2.2** ➠	**4.2.2** ➠
	4.2.3 Obtain input from the program user.	**4.2.3** Validate user input.	**4.2.3** ➠
	4.2.4 Identify truncation and round-off errors related to data types.	**4.2.4** ➠	**4.2.4** ➠
	4.2.5 Format numeric and textual output.	**4.2.5** ➠	**4.2.5** ➠
	4.2.6 Use string data for input, concatenation, and output of text.	**4.2.6** Manipulate text, including extraction, concatenation, and comparison.	**4.2.6** ➠
	4.2.7 Describe the relationship between each data type and the amount of memory it uses.	**4.2.7** Describe how an array is stored internally and how an individual element is accessed.	**4.2.7** Describe how a two-dimensional array is stored internally.

Teacher Notes

(continued...)

General Learning Outcome 4 (continued)

Technology: Students will develop the abilities to use, manage, and understand information and communication technologies by exploring programming languages and computer-controlled devices.

	Senior 2 Computer Science (20S) Students will...	Senior 3 Computer Science (30S) Students will...	Senior 4 Computer Science (40S) Students will...	Teacher Notes
SLO 4.2 Programming: Data Structures	**4.2.8** Use simple structures for the storage and retrieval of data.	**4.2.8** Use parallel arrays to store related data.	**4.2.8** Create a user-defined data structure (such as records) consisting of a combination of other data objects. **4.2.9** Create, store, retrieve, and manipulate sequential files and random-access files (such as database files and comma-delimited files).	
SLO 4.3 Programming: Control Structures	**4.3.1** Identify, use, and trace control structures, including – linear (sequential) – branching (if/then/else) – looping (definite and indefinite)	**4.3.1** Identify, use, and trace control structures, including – multiple branching beyond if/then/else (such as case, select case, and switch)	**4.3.1** Identify, use, and trace control structures, including – recursion	Looping can be used to create graphic animation.
SLO 4.4 Programming: Debugging	**4.4.1** Differentiate between compile-time errors, run-time errors, and logic errors. **4.4.2** Detect and correct compile-time errors, run-time errors, and logic errors. **4.4.3** Locate errors by hand tracing and by using debugging tools.	**4.4.1** Define sample data to test a program. **4.4.2** ➡ **4.4.3** ➡	**4.4.1** ➡ **4.4.2** ➡ **4.4.3** ➡	Debugging is a part of each programming assignment. Students should test each other's programs. In team projects, students should ask for assistance from other team members before requesting teacher assistance. This is intended to improve students' communication and problem-solving skills.

(continued...)

General Learning Outcome 4 (continued)

Technology: Students will develop the abilities to use, manage, and understand information and communication technologies by exploring programming languages and computer-controlled devices.

	Senior 2 Computer Science (20S)	Senior 3 Computer Science (30S)	Senior 4 Computer Science (40S)
	Students will...	Students will...	Students will...
SLO 4.5 Programming: Reusable Code	**4.5.1** Use existing code provided by the teacher to build a larger program.	**4.5.1** Create code that can be reused.	**4.5.1** Create code for other students to use within a larger group project.
			4.5.2 Define and use libraries.
SLO 4.6 Programming: Subprograms	**4.6.1** Use built-in functions with parameters, including graphic/drawing functions.	**4.6.1** ➠	**4.6.1** ➠
	4.6.2 Use and create subprograms that do not pass parameters.	**4.6.2** Use subprograms that pass parameters.	**4.6.2** Use and create subprograms that pass parameters.
		4.6.3 Recognize the scope of a variable (for example, differentiate between local and global variables).	**4.6.3** ➠

Teacher Notes

Facilitate the writing of more complex programs by providing pre-written segments of code that students may not yet understand. Students progress from using code, to planning to create reusable code, to sharing code with others.

(continued...)

General Learning Outcome 4 (continued)

Technology: Students will develop the abilities to use, manage, and understand information and communication technologies by exploring programming languages and computer-controlled devices.

	Senior 2 Computer Science (20S) Students will...	Senior 3 Computer Science (30S) Students will...	Senior 4 Computer Science (40S) Students will...	Teacher Notes
SLO 4.7 Programming: Objects		**4.7.1** Define the term *object*. **4.7.2** Use dot notation (for example, invoke a method or access a property).	**4.7.1** Define the term *class*. **4.7.2** Describe object-oriented programming in terms of encapsulation, inheritance, and polymorphism. **4.7.3** Create a program using objects and create classes using an object-oriented programming language. **4.7.4** Limit the scope of public and private identifiers appropriately.	The intent is to introduce the language of object-oriented programming early and to have students use objects as soon as possible. Students are not required to create their own classes until S4. The key features of object-oriented programming are: encapsulation, inheritance, and polymorphism. Not all languages support true object-oriented programming.
SLO 4.8 Programming: Algorithms		**4.8.1** Program simple algorithms (such as insertion sort and selection sort). **4.8.2** Compare the operation of algorithms. **4.8.3** Use search algorithms, including sequential and binary.	**4.8.1** Program advanced algorithms (such as quick sort, bisection search, and random number generator). **4.8.2** Compare the efficiency of algorithms. **4.8.3** ➠	The intent is to learn about algorithms, not to focus on programming sorts or searches.

Appendix

Suggested Course Outlines

The following chart provides an overview of suggested course outlines for Senior Years computer science courses.

Senior 2 Computer Science (20S)	Senior 3 Computer Science (30S)	Senior 4 Computer Science (40S)
This introductory course in computer programming is of interest to a broad audience. Students develop knowledge, skills, and attitudes applicable to situations beyond computer science.	This course takes an in-depth look at problem solving through computer programming.	This project-based computer programming course reflects the way projects are completed in industry. The course is structured to give students opportunities to write programs and develop project-management skills in a team-programming environment.
Topics:	**Topics:**	**Topics:**
• Strategies for independent learning, communication, and teamwork	• Strategies for independent learning, communication, and teamwork	• Strategies for independent learning, communication, and teamwork
• History of computer hardware	• History of computer software	• Current trends in ICT
• Impact of information and communication technologies (ICT) on society and the environment	• Impact of ICT on society and the environment	• Current ICT issues relating to society and the environment
• Ethics	• Ethics	• Ethics
• Exploring careers in ICT	• Researching careers in ICT	• Researching careers in ICT and presenting findings
• Programming standards and conventions	• Programming standards and conventions	• Future studies in computer science
		• Project management
		• Team project: create a computer simulation
		• Learning a new programming language independently
• Problem solving	• Problem solving	• Problem solving
• Programming concepts	• Programming concepts	• Programming concepts
– Documentation	– Documentation	– Sequential and random access files
– Variables and data types	– Variables and data types	– Writing a simple program using two different languages
– Input/output	– Input/output	– Object-oriented programming
– Debugging	– Debugging	– Recursion
– Branching (if/then/else), nested branching	– Multiple branching	– Arrays—two-dimensional
– Looping (definite and indefinite)	– Looping	– Analyzing algorithms
– Using functions/methods	– Subprograms with parameters	– Sorting and searching
– Built-in graphic functions	– Strings	
– Writing subprograms without parameters	– Arrays—one-dimensional, parallel	
	– Algorithms—simple sorts	

Notes

References

References

Manitoba Education and Training. *Computer Science 205*. Winnipeg, MB: Manitoba Education and Training, 1983.

---. *Computer Science 305*. Winnipeg, MB: Manitoba Education and Training, 1984.

---. *Technology As a Foundation Skill Area: A Journey Toward Information Technology Literacy: A Resource for Curriculum Developers, Teachers, and Administrators*. Winnipeg, MB: Manitoba Education and Training, 1998. Available online at <http://www.edu.gov.mb.ca/ks4/docs/support/tfs/index.html>.

Manitoba Education and Youth. *Subject Table Handbook: Student Records System and Professional School Personnel System*. Annual publication. Available online at <http://www.edu.gov.mb.ca/ks4/docs/policy/sth/index.html>.

Notes

Printed in Canada
Imprimé au Canada