

**Correlation of the textbook *Introduction to Programming in Turing* to the Ontario Computer and Information Science Policy Document for Grade 11 ICS3M**

<b>Expectation Code</b>	<b>Expectation</b>	<b>Page Number in textbook</b>
<b>Theory and Foundation</b>		
<b><i>Overall Expectations</i></b>		
TFV.01	- describe at least two problem-solving models;	pages 67-69, 216-217
TFV.02	- identify the stages in the software development process (problem definition, analysis, design, implementation, testing, and maintenance);	pages 62, 65-68
TFV.03	- explain standard control and data structures used in computer programs;	pages 11, 208-214, 222
TFV.04	- identify on-line and off-line resources;	page 67
TFV.05I	- explain the functions of basic computer components.	pages 14-17, 20-22, 24-25
<b><i>Specific Expectations</i></b>		
<b>Problem Solving, Logic, and Design</b>		
TF1.01	- define problems by identifying the expected output and necessary user input;	page 216
TF1.02	- explain how clarity at each step in the problem-solving process determines the quality and effectiveness of the final product;	pages 67-69, 216-217
TF1.03	- describe the steps in the software development process and their importance in the development of large programs or applications;	pages 62, 65-68
TF1.04	- explain different problem-solving models (e.g., top-down, bottom-up) that can be used to create a computer program;	pages 67-69, 216-217

TF1.05	- determine the level of error checking required for given problems;	pages 62, 67-68, 85-86
TF1.06	- identify the possibilities and limitations of proposed designs;	page 65-68
TF1.07I	- document for the user the potential and limitations of programs.	pages 67
Programming Concepts		
TF2.01	- describe the characteristics of integer, real, character, and Boolean data types;	pages 82-87, 230-231
TF2.02	- define constants, variables, expressions, and assignment statements;	pages 75, 82-83, 87-90
TF2.03	- describe the order in which arithmetic and logical operations are performed;	pages 75, 87, 111, Appendix E
TF2.04	- explain the use of Boolean operators in compound expressions;	page 111
TF2.05	- define the structure of one- and two-dimensional arrays and associated concepts (e.g., subscripts, elements, bounds);	pages 222-224, 231-233
TF2.06	- explain the purpose of selection and repetition structures, and how they are expressed in a programming language;	pages 11, 96-111, 152-162
TF2.07	- describe the purpose of functions and procedures, and how they are expressed in a programming language;	pages 254, 257-267, 269-270
TF2.08	- describe parameter passing and scope;	pages 254, 257-267, 276-284
TF2.09	- identify differences between local and global variables;	pages 263, 266, 284
TF2.10	- identify differences among logic, runtime, and syntax errors;	pages 13, 46, 85-86, 181-182, 186
TF2.11	- describe the evolution of programming languages (e.g., machine, assembly, high-level, 4GL);	pages 6-13
TF2.12	- evaluate available on-line resources such as "readme" files, help files, and "frequently asked questions" files;	page 67

TF2.13	- evaluate available off-line resources such as user manuals and reference manuals;	page 67
TF2.14	- explain the importance of external and internal documentation and programming style;	pages 62-67, 90-92, 106
TF2.15	- identify common acronyms used in the computing industry.	pages 4-28
Hardware, Interfaces, and Networking Systems		
TF3.01	- describe the function and location of the basic components of a computer (e.g., motherboard, CPU, I/O devices, memory);	pages 14-17, 20-22, 24-25
TF3.02	- identify common computer peripheral devices (e.g., mouse, keyboard, screen, printer, multimedia devices) and their primary functions;	pages 14, 17, 20-21, 26
TF3.03	- explain differences among software for systems, applications, and programming.	pages 25-28
<b>Skills and Processes</b>		
<b><i>Overall Expectations</i></b>		
SPV.01	- develop effective programs by following the steps in the software design process;	pages 65-68
SPV.02	- use defined programming practices (e.g., headers, indentation, internal documentation, informative variable names);	pages 62, 64-67, 90-92, 106 (+ examples throughout)
SPV.03	- produce appropriate internal and external documentation;	pages 62, 64-65, 67, 91 (+ examples throughout)
SPV.04	- properly use an operating system, including a network.	pages 22-23, 26
<b><i>Specific Expectations</i></b>		
Problem Solving, Logic and Design		
SP1.01	- resolve ambiguities and missing information in problem statements;	pages 66, 216 (+ examples throughout)
SP1.02	- use the input, process, and output model to solve problems:	pages 82-90

	solve problems;	(+ examples throughout)
SP1.03	- select suitable data structures to represent information;	pages 222, 224-225
SP1.04	- develop and maintain a project plan that covers all aspects of the development process for a computer program;	pages 65-69, 216-217
SP1.05	- develop appropriate algorithms in text or diagram form to solve problems and verify solutions;	pages 67-69, 210, 216-217
SP1.06	- produce user-friendly input and output forms;	pages 40, 75-76, 216 (+ examples throughout)
SP1.07	- solve the same problem using various tools (e.g., a calculator and a computer program, a sort program and a spreadsheet/ database/ word processor sort function);	pages 26-28, 73-77, 103 ...
SP1.08	- verify solutions to problems;	pages 67-68
SP1.09	- incorporate modularity into the design process.	pages 11, 67, 217
Programming Practices		
SP2.01	- use constants, variables, expressions, and assignment statements to store and manipulate numeric, character, and logical data in programs;	pages 82-90, 230-231
SP2.02	- incorporate one-dimensional and two-dimensional arrays into computer programs;	pages 222-227, 231-233
SP2.03	- write programs that use related arrays to store and extract data;	pages 227-229
SP2.04	- use selection structures, counted and conditional loops, and nested selection and loop structures;	pages 96-110, 152-162, 209-214
SP2.05	- manipulate numbers and text using built-in subroutines;	pages 254-257, 268-269, 404, Appendix B
SP2.06	- write subroutines that pass parameters and use local and global variables;	pages 257-267, 267-270, 276-287

SP2.07	- implement a program design using sequence, selection, and repetition structures;	pages 72-77, 82-90, 96-111, 152-162...
SP2.08	- use on-line and off-line reference materials effectively;	page 67
SP2.09	- adhere to defined programming style, including naming conventions for variables and subroutines, indentation, and spacing;	pages 64, 66-67, 90, 92, 106, 254-292
SP2.10	- incorporate and maintain internal documentation to a specific set of standards, including author, date, file name, purpose, and explanatory comments of major statement groups;	pages 62, 64-65, 91 (+ examples throughout)
SP2.11	- develop external documentation (including pseudocode, diagrams, and charts) to summarize the design;	pages 67-69
SP2.12	- test completed programs with a full range of valid data to ensure that all components work as expected;	pages 62, 67
SP2.13	- interpret errors during testing and program execution;	pages 45-46, 85-86, 79, 186
SP2.14	- trace program execution using manual methods and software debugging tools;	pages 45-46, 179, 181-182
SP2.15	- identify and correct logic, runtime, and syntax errors in programs;	pages 13, 46, 85-86, 106, 181-182, 186
SP2.16	- use linear searches and simple sort routines in programs;	pages 226-227, 277, 306-307, 285-286, 300-302, 322-327
SP2.17	- write programs that access sequential files;	pages 198-203
SP2.18	- perform peer evaluations of internal documentation and programming style.	pages 62, 64-67, 90-92, 106, 254-273, 417 (+ examples throughout)
Hardware, Interfaces, and Networking Systems		
SP3.01	- use an operating system to perform tasks such as managing files and configuring	page 26

	hardware;	
SP3.02	- use built-in networking functions such as shared files and input/output devices;	pages 22-23
SP3.03	- use common Internetworking services to access and navigate global information resources;	pages 28-29
SP3.04	- develop computer resources to share information globally or locally;	pages 22-23, 28-29
SP3.05	implement a comprehensive backup strategy for files.	pages 43-44, 161
<b>Impact and Consequences</b>		
<i>Overall Expectations</i>		
ICV.01	- explain issues related to the ethical use of computers;	pages 30-33
ICV.02	- describe emergent technologies and their impact on society;	pages 29-33
ICV.03	- identify information systems and computer science career paths, and their educational requirements.	pages 62-63
<i>Specific Expectations</i>		
Effects of Information Technology		
IC1.01	- explain how the pervasiveness of computer technology affects daily life;	pages 29-33
IC1.02	- describe how information is gathered using computers and how this can affect peoples' privacy and right to information;	pages 30-31
IC1.03	- identify a number of available sources of career and educational information using networks and evaluate their reliability and accuracy;	pages 62-63
IC1.04	- describe, using presentation software, emergent technologies and their potential influence on society;	pages 29-33
IC1.05	- use appropriate strategies to avoid potential health and safety problems associated with computer use, such as	

	musculo-skeletal disorders and eye strain;	
IC1.06	- explain the importance to identifying career paths of keeping up to date on current articles and thought on computer technology.	pages 62-63
Postsecondary Education and Career Opportunities		
IC2.01	- identify postsecondary educational opportunities leading to careers in information systems and computer science, and report on their entry requirements;	
IC2.02	- identify which careers require computer expertise, using local or national media;	pages 29, 62-63
IC2.03	- identify opportunities for apprenticeship training and co-op programs.	pages