

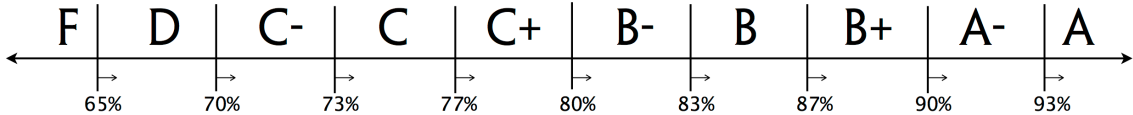
# Introduction to Programming

CMPT 120 • Fall 2014

## -Background

When and where	<i>Class:</i> Mondays and Thursdays at 2pm in Hancock 1021 <i>Lab:</i> Wednesdays at 9:30am in Hancock 0005	
Recommended Texts	<i>Internet and WWW How to Program</i> by Dietel, Dietel, & Dietel ISBN 978-0132151009 (table of contents at <a href="http://www.deitel.com/bookresources/iw3http5/toc.pdf">http://www.deitel.com/bookresources/iw3http5/toc.pdf</a> ) <i>Eloquent JavaScript</i> by Marijn Haverbeke available online at <a href="http://eloquentjavascript.net">eloquentjavascript.net</a>	
Optional Texts	<i>Code Complete, 2nd edition</i> by Steve McConnell ISBN 978-0735619678 <i>JavaScript: The Good Parts</i> by Doug Crockford ISBN 978-0596517748	
Web site	<a href="http://www.labouseur.com/courses/intro/">http://www.labouseur.com/courses/intro/</a>	
Instructor	Alan G. Labouseur Hancock 3017 (Office hours are posted.)	Alan.Labouseur@Marist.edu 845-575-3832 <i>Marist phone</i> 845-440-1102 <i>home office phone</i>

## -Grading

Letter Grades					
You can earn up to 1000 points over the course of the semester, broken down as follows: (These weights are subject to minor variation.)	Programming Projects	30.0%	300 points - five at 60 points each	[1,2]	
	Final Project	20.0%	200 points - v 1.0 of your game	[1,2]	
	Mid-term Exam	20.0%	200 points - one-page study sheet	[5]	
	Final Exam	20.0%	200 points - no study sheet	[5]	
	Attendance	2.5%	25 points	[1]	
	Constructive Participation	2.5%	25 points	[1]	
	Laziness Adjustment	2.5%	25 points - for <b>not</b> being lazy	[1]	
	Whining Adjustment	2.5%	25 points - for <b>not</b> whining	[1]	

## -Objectives and Assessment

Assessment methods include assignments, quizzes, exams, discussions, presentations, peer review, and projects.

[References] refer to Department of Computing Technology Goals available at <http://www.labouseur.com/courses/goals.pdf>.

This course introduces a **disciplined approach to the craft of software development**. Students learn to design, develop, test, debug, and document a program with good code style. This helps to form in the student a foundation for further studies in computer science. The students will:

- come to know software development as both art and science [1, 2a]
- understand how data is represented in a computer [2a, 2b]
- develop a mastery of basic Web markup and presentation technologies [1]
- come to know and use correctly data types, operators, and control structures [2a]
- be able to correctly use selection and repetition control structures [2a]
- believe in the wisdom of encapsulated objects [2e]
- be able to design and implement simple classes for problem solving [2a]
- be able to declare and manipulate arrays in at least two dimensions [2a]
- get practice in finding some answers for themselves, because capable problem solvers never stop learning. [1, 2a]

# Introduction to Programming

CMPT 120 • Fall 2014

## Proposed Schedule

#	Chapters	Topic	To Do
1	2, 4, 6	Monday 9/1 No class meeting - Labor Day Thursday 9/4 Interactive Fiction / Simple HTML and CSS / Document Object Model (DOM)	Read the books
2	2 12, 13	Monday 9/8 Course Expectations and Goals / a little Git / More HTML, CSS, and DOM Thursday 9/11 JavaScript / <b>Sequence</b> / "wiring up" events / more Git / Text Areas / Memory	Read the books
3	6.4-5, 7.11 7.5, 7.6	Monday 9/15 Code structure / Variables and declarations / (re)Assignment and Memory Thursday 9/18 <b>Alternation</b> with IF-ELSE / Data Types / <b>Software Development Best Practices</b>	<b>Project 1 due 9/18</b>
4	6.5, 9.1 - 9.4 -	Monday 9/22 Functions and Scope / Parameter passing / The gory details of memory Thursday 9/25 No class meeting - Alan is out of town	Read the books
5	9.7 - 9.8 -	Monday 9/29 Functions and Scope / Parameter passing / Memory / Modifying the DOM Thursday 10/2 Guest Speaker Vinnie Amatulli from Goldman Sachs on entrepreneurship	<b>Project 2 due 10/2</b>
6	7.7, 6.6, 8.3, 8.5, 8.8	Monday 10/6 Revisit <b>Alternation</b> , but now with SWITCH-CASE / Writing modular code Thursday 10/9 More on modular code and function specialization / Relational && logical ops	Read the books
7	- -	Monday 10/13 <b>Mid-term Exam part one</b> , in class, with a study sheet Thursday 10/16 <b>Mid-term Exam part two</b> , in class, with a study sheet	Read the books
8	9.2-3	Monday 10/20 Guest Speaker Alan Rothschild from Ipreo on professional-level programming Thursday 10/23 Debrief the Mid-term Exam results and review / Review project 4 requirements	<b>Project 3 due 10/23</b>
9	11	Monday 10/27 <b>Repetition</b> with FOR and WHILE loops / more Relational && logical ops Thursday 10/30 External JavaScript / Programming with objects	Read the books
10	11	Monday 11/3 Encapsulation via Object-oriented programming / Bases 2, 10, 16 / more CSS Thursday 11/6 Prototypes and object trees / Software Development Best Practices	Read the books
11	10	Monday 11/10 Guest Speaker Rob Fagnoni from Morgan Stanley on resumes and interviewing Thursday 11/13 Guest Speaker Dan Miller from Etsy / OOP: inheritance	<b>Project 4 due 11/13</b>
12	9.5, 10 3, 14	Monday 11/17 Arrays / Iterating over arrays / Arrays of (other) objects Thursday 11/20 Arrays of Arrays / initializing / Arrays as parameters	Read the books
13	- -	Monday 11/24 Modern HTML5 input and semantic elements Thursday 11/27 No class meeting - Thanksgiving Break	<b>Project 5 due 11/24</b>
14	- 9.9, 9.10	Monday 12/1 Matrices (2D arrays) as navigation for our game Thursday 12/4 JavaScript functions as first-class objects / Functional programming notions	Read the books
15	- -	Monday 12/8 Recursion and dain bramage / Iterative vs. Recursive solutions - Fibonacci Thursday 12/11 Review for Final Exam	<b>Final Project due 12/11</b>
16	-	Monday 12/15 at 1pm <b>Comprehensive Final Exam</b> No study sheet. No books. No notes. Just you, a pencil, and some paper.	Reflect on having read the books