LaSalle College
Montréal

COURSE OUTLINE

Course identification

Name of programs – Codes:	COMPUTER SCIENCE TECHNOLOGY - PROGRAMMING - 420.BP INFORMATION TECHNOLOGY PROGRAMMING - LEA.3Q
Course title:	**WEB CLIENT DEVELOPMENT I**
Course number:	420-WC1-AS
Total number of course hours:	90 hours
Weighting:	3-3-4
Statement of the competencies – Codes:	Use of productivity software. – 00Q4 Develop non-transactional Web applications – 00ST

Contribution of the course in the program

Course position

This course is offered in the first semester of the programs *Computer Science Technology – Programming* (420.BP) and *Information Technology Programming* (LEA.3Q). Its duration is 90 hours, divided into 45 hours of theory and 45 hours of practical exercises, plus approximately 60 hours of personal work (assignments). It shares the development of the 00Q4 competency with *UML Modeling* (420-UML-AS) in the third semester. It also shares the competency 00ST with *Web Client Development II* (420-WC2-AS) in the second semester. This course has no prerequisites. It is a prerequisite to *Web Client Development II* (420-WC2-AS) in the second semester, *Web Server Application Development I* (420-WS1-AS) in the fourth semester and *Web Server Application Development II* (420-WS2-AS) in the fifth semester.

Scope of the course

In this course, we offer a direct and engaging immersion into the fundamentals of modern coding and design, transforming the student from a simple user into a true Web artisan. Capable of designing professional interfaces from A to Z according to the highest industry standards, the student adopts a “Code-First” approach focused on manual coding. The program ensures an in-depth technical mastery of semantic HTML5 and CSS3, enabling the construction of robust structures through expert use of Flexbox, CSS Grid, and relative units. Learning takes place within a modern professional workflow integrating the VS Code editor, version control with Git and GitHub, and a smooth transition from design to code using prototyping tools like Figma and Canva.

Aligned with sustainable development goals, the course emphasizes software eco-design by teaching resource optimization and code performance. Regarding artificial intelligence, its use is strictly limited and supervised, serving only as a support in certain learning activities to facilitate the understanding of complex concepts without replacing logical reasoning.

Upon completion of this course, the student will not only be able to structure, style, and deploy high-performance static and responsive websites, but will also have developed critical and versatile expertise in using essential frameworks such as Bootstrap. This comprehensive training enables full autonomy and provides a decisive competitive advantage for future internships in the industry.

Course components (objective and standard of the competencies)

Expected outcomes (achievement context of the competencies)

The achievement context of these competencies will reflect the conditions as they occur in the following settings: academic, professional, work, or life environment.

- **Achievement context – Competency 00Q4**
  - Using word processing software, spreadsheet software, design software, presentation software and collaborative software
  - Using images, sounds and videos
  - Using presentation standards
- **Achievement context – Competency 00ST**
  - For Web applications associated with information delivery, marketing, etc.
  - For new applications and applications to be modified
  - Based on design documents
  - Using images
  - Using issue tracking and version control procedures

Throughout the course, you will engage in various learning situations/activities so that by the end of the course, you will have met the expected outcomes.

Elements and performance criteria

The elements of an objective formulated in terms of the competency specify its essential components. They include only what is necessary in order to understand and master the competency. If the competency is described as a process, the elements are the steps for execution.

The performance criteria are the specific pre-established requirements upon which you and your teacher can objectively judge your development of the targeted competency. They are part of the description of this competency. They are prescriptive.

Sometimes an element appears in more than one course. If this is the case, a number indicates its complexity level: level one (1) being the simplest, level two (2), average, and level three (3), advanced, at the ministerial level.

Below are the elements of the competencies and performance criteria for this course that are to be respected:

Competency: Use of productivity software. – 00Q4

General ministerial and institutional performance criteria:

– N/A

Elements of the competency	***Performance criteria specific to each element***
4. Produce presentation documents.	4.1 Proper customizing of the presentation software interface 4.2 Appropriate choice of the display resolution and format 4.3 Appropriate integration of images, sounds and videos 4.4 Presentation readability 4.5 Compliance with spelling and grammar rules 4.6 Compliance with presentation standards
5. Share and synchronize documents.	5.1 Proper customizing of the collaborative software interface 5.2 Appropriate conversion of file formats 5.3 Appropriate classification of documents 5.4 Correct assignment of access to shared documents 5.5 Efficient management of conflicts between versions

Competency: Develop non-transactional Web applications – 00ST

General ministerial and institutional performance criteria:

– N/A

Elements of the competency	***Performance criteria specific to each element***
4. Program the Web interface.	4.1 Appropriate use of markup language 4.2 Suitable creation and use of style sheets 4.3 Proper integration of images 4.4 Adaptation of the interface based on the display format and resolution
6. Program the client-side application logic.	6.1 Correct manipulation of DOM objects 6.2 Proper programming of interactions between the Web interface and the user

	6.3 Proper programming and integration of animations and widgets
7. Control the quality of the application.	7.1 Precise application of test plans 7.2 Thorough reviews of code and security 7.3 Relevance of the corrective actions 7.4 Compliance with issue tracking and version control procedures 7.5 Compliance with design documents

Course content/main themes

*Listed below is the **essential** content to be covered in this course:*

Chapter 1: Modern Web Ecosystem, Communication Protocols, and Development Environment Setup

- Infrastructure and protocols: Understanding the request/response cycle (HTTP/HTTPS), status codes (200, 404, 500), and the role of servers and DNS
- Professional environment: Setting up VS Code and DevTools, along with an early introduction to DevOps culture using Git and GitHub

Chapter 2: Information Architecture with Semantic HTML5, Accessibility, and SEO Optimization

- Semantics and structure: Organizing content using structural tags (header, main, footer) to optimize accessibility and SEO
- Forms and interactivity: Designing valid and well-structured forms to ensure data input meets current standards

Chapter 3: Systematic Design with CSS3, Flow Management, and Fundamental Visual Styling Principles

- Foundations and Box Model: Mastery of selectors, typography, and the box model to move beyond ad hoc styling
- Variables and consistency: Using custom properties (CSS variables) to ensure maintainability and design consistency

Chapter 4: Advanced Layout Mastery with Flexbox, CSS Grid, and Relative Units

- Modern layout: In-depth use of Flexbox and introduction to CSS Grid for creating complex and flexible layouts
- Dynamic sizing: Precise use of relative units (rem, em, %, vw, vh) for fluid interfaces

Chapter 5: Responsive Design Strategies and Mobile-First Approach

- Adaptive design: Implementing media queries to adjust interfaces based on screen resolution
- Mobile-first methodology: Prioritizing content and structure for small screens before scaling to larger ones

Chapter 6: Interface Design with Prototyping Tools and Design-to-Code Transition

- UI/UX prototyping: Creating wireframes and high-fidelity mockups using tools like Figma or Canva
- Design-to-code integration: Methodology for translating visual mockups into functional HTML/CSS structures

Chapter 7: Modern CSS Frameworks: Component-Based Approach (Bootstrap)

- Component framework (Bootstrap): Using grids and pre-styled components (Navbar, Cards, Modals) for rapid prototyping
- Optimization and performance: Managing code weight and applying cleanup principles (e.g., PurgeCSS) for high-performance websites

Chapter 8: Collaborative Workflow Management, Continuous Deployment, and Professional Publishing

- Full GitHub workflow: Advanced collaboration, branch management, and project hosting
- Deployment and publishing: Free and automated deployment of static websites via GitHub Pages or Netlify for immediate public access

Learning activities

Provided below are examples of learning activities that correspond to the competencies for this course. The learning activities are found in the course calendar that complements this course outline.

- **Manual Coding Labs**: Hands-on exercises writing semantic HTML5 and CSS3 without the aid of generators to master the deep structure of the DOM.
- **"Design-to-Code" Challenges**: Technical workshops focused on converting mockups created in Figma or Canva into functional websites that faithfully replicate the original design.
- **Responsive Layout Projects**: Creation of adaptive interfaces using Flexbox and Media Queries following a "Mobile-First" methodology.
- **Collaborative Versioning Workshops**: Systematic use of Git and GitHub to archive work, simulating a real professional development environment.
- **Framework Immersion**: Practical comparison of rapid component-based integration (Bootstrap).
- **Capstone Project**: End-of-term project involving the full design and deployment of a multipage website, including initial prototyping, responsive development, and public launch.

Terms for Evaluating Learning

The evaluation of your learning is based on two inseparable methods: formative evaluation and summative evaluation. These two evaluation types are formal. Detailed information on the evaluation schedule is found in the course calendar, under the "Formative and summative evaluation schedule" column.

Formative evaluation

Following a learning activity or learning period, time is set aside for introspection. You will determine what has been understood and achieved and seek to identify the nature and origin of weak areas. These designated periods consist of simple means: short tests, association games, logbooks, a portfolio, questions, creating of samples, etc.

Formative evaluation is frequent and covers as many aspects as possible. It takes place in class, individually or in groups, and leads to immediate decisions. ***You are the one who assumes the bulk of the work during individual or group corrections, adjustments and other self-evaluation tasks. The purpose is not to determine grades.***

If you take the results of the formative evaluations seriously throughout the course, you will ensure preparedness for the summative evaluations. You will be able to make the necessary progress to acquire the targeted competency at the required level, according to the achievement context and pre-established performance criteria.

Below are some examples of formative evaluation methods that correspond to the targeted competencies for this course:

- **Peer Technical Reviews on GitHub**: Exchange of code comments to validate HTML semantics and CSS cleanliness.
- **Timed Layout Challenges**: Quick integration exercises (e.g., a navigation bar with Flexbox) to assess speed and mastery of workflow.
- **Self-Evaluation with DevTools**: Use of browser development tools to inspect and correct box model or responsive design errors in real time.
- **Syntax and Protocol Quizzes**: Short tests on HTTP status codes, CSS selectors, and semantic hierarchy to reinforce theoretical foundations.
- **Design Critique and Prototyping**: Presentation of Figma or Canva wireframes before coding to validate usability and intended visual structure.
- **W3C Validation**: Systematic submission of files to the validator to ensure code compliance with current standards.

Summative evaluation

Summative evaluations are less frequent. They take place later on, towards the middle and end of the semester. This gives you the time to integrate your learning and to learn how to apply it to situations related to the targeted competency. The summative evaluation material is prepared by your teacher according to the description of the course's targeted competency: its elements, achievement context and performance criteria.

The work completed in summative evaluations is graded. The purpose is to determine what you have learned.

Below is the information on the summative evaluation schedule and details for this course, as well as the weighting of marks:

Evaluations	***Weighting***
Mid-term exam	30%
Project	30%
Final exam	40%
Total	100%

Institutional requirements

Student's commitment

By registering for this course, you commit to:

- *obtain the necessary course materials at the start of the semester;*
- *respect the copyright;*
- *participate in the learning activities, formative and summative evaluation activities outlined in the course calendar;*
- *complete the work assigned to you;*
- *submit the work on time.*

Here are some particular commitments for this class:

- *Syntax Precision: Commit to writing clean, valid code manually, avoiding automated solutions to fully master the mechanics of the Web.*
- *Collaborative Culture: Systematically use Git and GitHub to archive work and learn to operate according to professional industry standards.*
- *Technological Curiosity: Actively explore the differences between component-based approaches (Bootstrap) and utility-based frameworks (Tailwind CSS) to develop critical judgment.*
- *Design-First Methodology: Follow a structured workflow by always creating a prototype in Figma or Canva before starting technical implementation.*
- *Autonomy and Problem Solving: Learn to use development tools (DevTools) to diagnose and fix your own layout errors.*

Teacher's commitment

Your teacher commits to:

- *create varied learning situations that enable you to put into practice the knowledge, actions and professional behaviour of the targeted competency;*
- *plan sufficient and appropriate formative evaluation activities, involving correction and improvement, that provide frequent feedback, allowing you to be well informed of your progress;*
- *provide summative evaluations that correspond to the course's targeted competency;*
- *evaluate work according to the applicable criteria, in a fair and equitable manner within a reasonable time.*

Here are some particular commitments for this class:

- *Technological Updates: Ensure instruction is based on current industry standards, replacing outdated tools with robust core technologies.*
- *Hands-On Guidance: Support students in setting up and mastering their professional environment, including VS Code and Git.*
- *Design Relevance: Reinforce the "Design-to-Code" workflow by integrating modern tools like Figma and Canva to simulate real-world market practices.*
- *Responsive Guidance: Facilitate learning of adaptive layout techniques and the "Mobile-First" approach.*
- *Debugging Support: Provide technical assistance during labs to help students understand the DOM hierarchy and the mechanics of the Web.*

The Institutional Policy on Evaluating Learning (IPEL) is applied to all institutional programs. Listed below are a few of its clauses:

Written language *(article 5.7)*

The teacher is responsible for identifying spelling and grammar errors and for allocating the corresponding number of marks for any given summative evaluation.

Below is the % – based on language requirements – that can be attributed to each summative evaluation:

- Up to 10%

Class attendance *(article 5.12)*

Attendance and participation in classes and evaluations are mandatory for all students.

The teacher has the responsibility of monitoring attendance and of evaluating the reasons justifying student absences from classes.

A student whose absences exceed the allowable number for the course could be denied access to the final exam for that course.

Plagiarism and fraud *(article 5.16)*

Plagiarism, attempted plagiarism or complicity in plagiarism during an assignment or any evaluated task contravenes the rules. This includes (but is not limited to):

- *the whole or partial presentation (reference, paraphrase, summary, translation, insertion) of the work of another (text, illustration, film, music, etc. on paper or online) as one's own, or failing to cite a source;*
- *the use of another student's exam during an exam;*
- *the use of an assignment done for another course or a project already submitted in the past, which is passed off as an original work.*

Fraud, attempted fraud or complicity in fraud constitutes an infraction.
This includes (but is not limited to):

- *the possession or use of any unauthorized document, material or equipment during an exam, including the use of technological tools;*
- *the execution of an evaluated task by another person;*
- *the substitution for another person during an exam, assignment or any evaluated task;*
- *the possession of the questions or answers of the exam;*
- *the obtainment of any aid not authorized in advance by the teacher.*

Plagiarism, attempts at plagiarism or fraud, or collaboration in plagiarism or fraud are prohibited and considered serious offences. Thus, any instances of plagiarism or fraud will lead to a grade of '0' for the assignment in question. In addition, a note will be made in the student's file and the student will receive a written notice from his or her Program Directorate to that effect.

In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question. A second note is made in the student's file and the student will receive a summons from his or her Program Directorate. For a third offence, he or she may be expelled from the College.

Submission of work and tests *(article 5.8)*

All assignments must be submitted in class at the time designated by the teacher. Any late submissions result in a grade of zero (0).

Upon presentation of an official supporting document or valid reason for the absence, the student may request an extension from the teacher, who may accept or refuse the student's work and apply a penalty for the lateness.

Program Directorates do not accept student work. Assignments must be submitted directly to the teacher.

Rules and regulations to follow

Late arrivals

The teacher may refuse to admit to the classroom any student arriving late. A late arrival is considered an absence for that period.

Note: Students arriving late must recognize that the information they missed will not be repeated. Late students are therefore responsible for asking their peers about the material they missed. Arriving after the break, as well as leaving before the end of the class, may result in one or more hours of absence.

Eating and drinking in class

Eating and drinking are prohibited in the classrooms, locker rooms and Documentation Centre. Food may only be eaten in the cafeteria, vending machine areas and student lounges.

Mandatory course material

- Laptop with specifications mentioned on the college's website. LaSalle College. Bring Your Own Device. 2017. http://www.lasallecollege.com/futurstudents/bring-your-own-device
- Professional Code Editor: Free installation of Visual Studio Code with extensions recommended by the instructor.
- Version Control Tools: Installation of Git (Git Bash for Windows) and creation of a GitHub account.
- Modern Web Browser: Use of Google Chrome or Firefox to access built-in development tools (DevTools).
- Design Platforms: Access to online prototyping tools such as Figma and Canva for the graphic design phase.
- Deployment Environment: Access to static site hosting services like GitHub Pages or Netlify.

Bibliography for this course

BOOTSTRAP OFFICIAL DOCUMENTATION: *Comprehensive Guide for Using the Grid System and Assembling Interfaces with Pre-Styled Components*. https://getbootstrap.com/docs/

MDN WEB DOCS (Mozilla): *Resources for Developers, by Developers*. https://developer.mozilla.org/en-US /

Academic Studies Directorate approval:

Emmanouela Tisizi pour la DÉ

Signature numérique de Emmanouela Tisizi pour la DÉ
Date : 2026.03.31 13:19:46 -04'00'