



# AMERICAN COLLEGE

## CSC414 COMPUTER SCIENCE PROJECT I

### ECTS Course Syllabus

<b>Code</b> CSC414	<b>Title</b> Computer Science Project I	
<b>ECTS Credits</b> 12	<b>Department</b> Computer Science	<b>Instructor</b>
<b>Semester</b> Fall, Spring	<b>Cycle</b> First	<b>Language of Instruction</b> English

### Description

The Computer Science Project I is the first part of a capstone project for the senior year. Students during this course and CSC415 Computer Science Project II complete an individualized learning experience. Beginning with a driving question, students will research, work with an academic supervisor and complete a project, which will then officially present to their supervisor. This course will showcase most of the techniques and experience students have gained while studying their BSc in Computer Science.

### Learning outcomes

By the end of the course, students are expected to:

- apply the theoretical knowledge gained in previous computer science courses in order to design a software product;
- perform all the necessary steps in order to conduct feasibility analysis and gather the requirements for the system to be created;
- have the knowledge, skills, and attitudes to become: self-directed, lifelong learners, flexible workers complex thinkers and effective communicators;
- demonstrate organizational, time management and communication skills;
- critically examine, investigate and prepare project reports; and
- demonstrate the ability to use Standard English grammar, punctuation, spelling and usage.

**Prerequisite(s):** Completion of 180 ECTS credits

### Learning methods and educational activities

Meetings, project proposal preparation, discussions, research activity, writing project work, independent and private study.

**Teaching hours:** 0

### Assessment methods and weight

Project:	100%
----------	------

### Grading system

90-100	<b>A</b>	85-89	<b>B+</b>	80-84	<b>B</b>	75-79	<b>C+</b>
70-74	<b>C</b>	65-69	<b>D+</b>	60-64	<b>D</b>	0-59	<b>F</b>

### Content

A project in the area of Computer Science is chosen by the student, subject to their supervisor's approval, who makes a literature review and performs the SDLC phases to fulfil the project. The project work is supervised by a member of the Computer Science department.

<b>Week 1</b>	<b>Critical choice of method, in the creation of a relevant theoretical frame of reference.</b>
<b>Week 2</b>	<b>Extensive analysis of the theoretical and empirical material in which the author makes a contribution to the literature in the chosen subject or problem area.</b>

<b>Week 3</b>	<b>Introduction:</b> This section contains a brief generic presentation of the project undertaken, its main phases, as well as an outline of the remaining document.
<b>Week 4</b>	<b>Critically reviewing the literature:</b> The critical review; literature sources available; planning the literature search; conducting the literature search; obtaining and evaluating the literature; recording the literature.
<b>Week 5-8</b>	<p><b>Deciding on the development approach and choosing a development strategy:</b></p> <p><b>Purpose and Scope of the Project</b></p> <ol style="list-style-type: none"> <li>Motivation The motivation behind the project is described (automation or upgrading of an existing information system, bespoke software, part of an overall hardware system).</li> <li>Existing System Description (Optional) If the software will replace an existing manual or semi-automatic information system, the elements of the existing system need to be described (organization, people, work, environment).</li> <li>Project scope A rough-cut description of the software is presented. Major inputs, processing functionality and outputs are described without regard to implementation detail.</li> <li>Performance/Behavior issues Any special requirements or conditions for performance or behavior are noted here.</li> <li>Management and technical constraints Any special constraints that affect the manner in which the project will be conducted, the technical approach to development, and any concerns that the developer might have regarding any system aspect are noted here. Included here should be any concerns that the developer might have regarding any system aspect</li> </ol> <p>2. Feasibility Study</p> <ol style="list-style-type: none"> <li>Financial Feasibility A cost-benefit analysis of the proposed software implementation is provided.</li> <li>Operational Feasibility This section illustrates how the proposed implementation will satisfy user and/or organizational objectives.</li> <li>Technical Feasibility A preliminary examination of available technology for the implementation of the project is conducted.</li> <li>Schedule Feasibility This section investigates if the estimated project completion time lies within the available time resources.</li> </ol>
<b>Week 9-13</b>	<p><b>Analysis &amp; Design</b></p> <p>1. <i>Requirements Elicitation Methodology</i> The methodology and techniques that were utilized for the elicitation of customer requirements are described in this section (interviews, brainstorming, scenario analysis, prototyping).</p> <p>2. <i>Software Requirements Specification</i></p> <ol style="list-style-type: none"> <li><i>Usage Scenarios</i> A number of usage scenarios are provided that describe the typical functionality of the software, as this was understood through the process of requirements elicitation. UML diagrams can be employed to assist the understanding of scenarios.</li> <li><i>Software modeling</i> <ol style="list-style-type: none"> <li>Database modeling A description of the database model that will support the implementation of the software is provided. This constitutes the Entity-Relationship (ER) diagram in a pictorial form, together with a textual description of its structure.</li> <li>Functional modeling A modeling tool such as Dataflow diagrams or UML should be used to provide a detailed description of the major software functions. The context diagram and Level-1 DFD diagrams are presented in a pictorial form and explained with the help of narrative text. The lower-level DFD diagrams are provided in the relevant Appendix section. In the case of Object-Oriented modeling, a description of the primary classes of the solution should be provided in this section. The reader should be referred to the relative Appendix for a complete and detailed listing.</li> <li>Behavioral modeling (Optional) The time-dependent behavior of the software is presented with the help of a State-Transition Diagram or a similar modeling tool. Both a pictorial representation and a (brief) textual description of the respective diagrams should be provided.</li> <li>System Architecture A pictorial representation of the software architecture in the form of a structure chart or an equivalent modeling tool is provided. If the software has been developed as a part of a larger system implementation, the architectural diagram of the system should be provided as well.</li> </ol> </li> </ol>

When grading the Computer Science project, the supervisor will consider a number of issues, including but not limited to:

- Quality, clarity, and thoroughness of proposal
- Quality and quantity of sculptural work
- The general professionalism shown by the student, this includes; scheduling of meetings, timeliness handing in proposal revisions, attitude towards committee and colleagues, and studio work habits and time management
- Mastery of materials
- Problem-solving abilities
- Ability to work independently
- The improvement made from the beginning of the semester.

### **Student workload**

<b>Activity</b>	<b>Hours</b>
Meetings	10
Independent Study and Research Activity	210
Development strategy	20
Analysis and Design	20
Project Report Preparation	40
<b>Total</b>	<b>300</b>