1 What the students should know and be able to do

1. Students should be able to
   
   (a) Know the notions of variables and expressions
   (b) Compute with numbers
   (c) Manipulate sequences (Strings, Lists, and Files)
   (d) Define functions and use call-by-value or call-by-reference
   (e) Use decision structures and boolean expressions
   (f) Use definite and indefinite loops
   (g) Define classes and use objects in OOP
   (h) Use recursion in algorithms
   (i) Apply all basic concepts to build a more complex project
2 Mapping to Attributes with their Indicators

A01 Knowledge
(3) Competence in Engineering Fundamentals
(4) Competence in specialized engineering knowledge

A02 Analysis
(5) Ability to identify the essential characteristics of a technical problem, including scope
(6) Ability to identify reasonable assumptions (including identification of uncertainties and imprecise information) that could or should be made before a solution path is proposed
(7) Ability to identify a range of suitable engineering fundamentals (including mathematical techniques) that would be potentially useful for analyzing a technical problem
(8) Ability to decompose and organize a problem into manageable sub-problems
(9) Ability to obtain substantiated conclusions as a results of a problem solution, including recognizing the limitations of the solutions
(28) The ability to use of modern/state of the art tools

A03 Investigation
(10) Able to recognize and discuss applicable theory knowledge base
(11) Capable of selecting appropriate model and methods and identify assumptions and constraints

A04 Design
(17) Recognizes and follows an engineering design process
(18) Recognizes and follows engineering design principles
(25) Properly documents and communicates processes and outcomes

A05 Tools
A06 Work
A07 Communication
(37) Demonstrates an ability to respond to technical and non-technical instructions and questions
(39) Demonstrates appropriate use of technical vocabulary

A08 Professionalism
A09 Impact
A10 Ethics
A11 Economics
A12 Learning
A13 Sustainability
3 Course work

The course work consists of 11 weekly labs (100%).

4 Prerequisites

No prerequisites.

5 Learning outcomes

<table>
<thead>
<tr>
<th>Topic</th>
<th>Below</th>
<th>Marginal</th>
<th>Meets</th>
<th>Exceeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>doesn’t know what variables and expressions are</td>
<td>has an incomplete grasp of variables and expressions</td>
<td>knows what variables and expressions are and how they are used</td>
<td>can identify complicated expressions</td>
</tr>
<tr>
<td>1b</td>
<td>doesn’t know how to use numbers in computation</td>
<td>can use some numerical expressions but has difficulties with different numerical types</td>
<td>can use the correct numerical operations for different numerical types</td>
<td>turns numerical data into their correct type before applying numerical operations</td>
</tr>
<tr>
<td>1c</td>
<td>doesn’t know how to manipulate sequences</td>
<td>knows how to manipulate some kinds but not all kinds of sequences</td>
<td>knows how to manipulate all kinds of sequences</td>
<td>can perform sophisticated manipulations of sequences</td>
</tr>
<tr>
<td>1d</td>
<td>cannot properly define functions</td>
<td>can define functions but cannot use properly call-by-value or call-by-reference</td>
<td>can define functions and use call-by-value or call-by-reference</td>
<td>can do structured and sophisticated function definitions and uses of call-by-value or call-by-reference</td>
</tr>
<tr>
<td>1c</td>
<td>cannot use decision structures and boolean expressions</td>
<td>can use some decision structures and boolean expressions</td>
<td>can use decision structures and boolean expressions</td>
<td>can make a sophisticated use of decision structures and boolean expressions</td>
</tr>
<tr>
<td>1d</td>
<td>cannot use definite and indefinite loops</td>
<td>can use some definite and indefinite loops</td>
<td>can use correctly definite and indefinite loops</td>
<td>can make a sophisticated use of definite and indefinite loops</td>
</tr>
<tr>
<td>1g</td>
<td>cannot properly define classes and use objects in OOP</td>
<td>can define classes and cannot use properly objects in OOP</td>
<td>can define classes and use objects in OOP</td>
<td>can do structured and sophisticated class definitions and uses of objects in OOP</td>
</tr>
<tr>
<td>1h</td>
<td>cannot use recursion in algorithms at all</td>
<td>can use some simple recursive definitions</td>
<td>can use recursion in algorithms</td>
<td>can make a sophisticated use of recursion in algorithms</td>
</tr>
<tr>
<td>1i</td>
<td>cannot apply basic concepts to build a more complex project</td>
<td>can apply basic concepts to partially build a more complex project</td>
<td>can apply basic concepts to build a complex project</td>
<td>can make a sophisticated application of all basic concepts to build a complex project</td>
</tr>
</tbody>
</table>