DeCAIR Course Syllabus Form

Author(s)	Naeem Al-Oudat		
Author Organization Name(s)	Tafila Technical University		
Work Package Number & Title	Work Package 2: Development of new MSc and BSc programs in AIR		
Activity Number & Title	Activity 2.2: Designing and developing syllabi and content for the agreed upon courses in the new programs		
Work Package Leader	Francesco Masulli, University of Genoa		
Due Date of Delivery	1/2/2022	**Project Month**	M14
Submission Date	4/11/2021	**Project Month**	M11

Revision History

Version	Date	Author	Description	Action *	Page(s)
1	4/11/2021	Naeem Al-Oudat	Original (base) document	C	1-5
2	10/1/2022	Murad Alaqtash	Revised Version	U	2-3
3	16/1/2022	Naeem Al-Oudat	Revised version based on a peer review	U	2-3
4	2/2/2022	Naeem Al-Oudat	Revised based on an expert review	U	2-3

(*) Action: C = Creation, I = Insert, U = Update, R = Replace, D = Delete

Disclaimer

This project has been co-funded by the Erasmus+ Programme of the European Union.

Email: DeCAIR@ju.edu.jo

Project Website: http://DeCAIR.ju.edu.jo/

Course title	Programming for Engineers
Course number	0107200
Credit hours (lecture and lab)	3 (3 + 0)
ECTS (weekly contact and self-study load)	6 (3 + 3)
Prerequisites/co-requisites by course number and name	None
Prerequisites by topic (other than the formal prerequisites above)	Students are assumed to have good background in basic algebra.
Level and type (compulsory, elective)	Bachelors' compulsory course
Year of study and semester	First year, first semester
Catalogue description	This course is a beginning programming course for engineering students that concentrates on the fundamentals of a programming language (C, C++, Java, Python, …) and coding principles. It emphasizes on basic topics such as data types, control flow, functions, arrays, strings, and dynamic memory allocation. In this course, students learn-by-doing how to write well-designed and structured computer programs to solve engineering problems.
Objectives	1. Introduce students to the basic concepts of writing codes by devising suitable algorithm for any engineering problem. 2. Introduce students to the modular programming, data structures and dynamic memory management. 3. Introduce students to the recursive programming techniques. 4. Emphasize the importance of following the conventional coding principles and apply debugging techniques.
Intended learning outcomes	Upon successful completion of this course, students will be able to:

No	Intended learning Outcome (ILO)	Program learning outcome (PLO)*
1	Conceptualize engineering problems as computational problems	1
2	Understand the fundamental software notations and coding principles	1
3	Handle the programming language (syntax and semantics) using IDE.	1
4	Write programs that are well-designed and structured	1
5	Write programs to solve engineering problems	1, 7

<table>
<tr><td></td><td>(*) The PLOs are listed in the appendix</td></tr>
<tr><td>Teaching and learning methods</td><td>Development of ILOs is promoted through the following teaching and learning methods:
Lectures will be delivered through Microsoft Teams and will be recorded for later access.
The AI lab is open for the students to practice the practical aspects and solve the programming homework assignments.
The student attends the class presentations and participates in the discussions.
The student joins the related online team/group and participates in its discussions.
The student studies the reference material, including books and videos.
The student solves the programming assignments in machine learning.
The student carries out a term project for solving a problem using ML techniques.
The student develops a professional report for the term report.
The student presents the term project in class.</td></tr>
<tr><td>Learning material type</td><td>Textbook, class handouts, some instructor keynotes, selected YouTube videos, and access to a personal computer and the internet.</td></tr>
<tr><td>Resources and references</td><td>A- Required book(s), assigned reading and audio-visuals:
Andrew Hilton and Anne Bracy. All of Programming, 1st edition, 2021. Online
B- Recommended book(s), material and media:
Lysecky, Roman, and Frank Vahid. Programming in C. ZyBooks. Online.</td></tr>
<tr><td>Topic outline and schedule</td><td>
<table>
<tr><th>Week</th><th>Topic</th><th>ILO</th><th>Resources</th></tr>
<tr><td>1</td><td rowspan="3">Introduction to programming</td><td rowspan="3">1</td><td rowspan="3">1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td rowspan="2">Reading code</td><td rowspan="2">1,2</td><td rowspan="2">1</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td><td>Language data types</td><td>2,3</td><td>1, 2</td></tr>
<tr><td>7</td><td>Writing code</td><td>2,3</td><td>1, 2</td></tr>
<tr><td rowspan="2">8</td><td>Compiling and running</td><td rowspan="2">2,3</td><td rowspan="2">2</td></tr>
<tr><td>Testing and debugging</td></tr>
<tr><td>9</td><td>Control flow</td><td>4</td><td>1,2</td></tr>
<tr><td>10</td><td>Functions</td><td>4</td><td>1,2</td></tr>
<tr><td>11</td><td>Arrays</td><td>4,5</td><td>1, 2</td></tr>
<tr><td>12</td><td>Strings</td><td>4,5</td><td>1, 2</td></tr>
<tr><td>13</td><td rowspan="2">Pointers and dynamic memory allocation</td><td rowspan="2">4,5</td><td rowspan="2">1, 2</td></tr>
<tr><td>14</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>15</td><td>Programming in large</td><td>4,5</td><td>1, 2</td></tr>
</table>
</td></tr>
<tr><td>Evaluation tools</td><td>Opportunities to demonstrate achievement of the ILOs are provided through the following assessment tools:
<table>
<tr><th>Assessment tool</th><th>Mark</th><th>Topic(s)</th><th>Time</th></tr>
<tr><td>Programming assignments</td><td>30%</td><td>Programming aspects</td><td>W2-W14</td></tr>
<tr><td>Midterm exam</td><td>30%</td><td>Introduction through classical techniques</td><td>W8</td></tr>
<tr><td>Final exam</td><td>40%</td><td>All material</td><td>W16</td></tr>
<tr><td>Total</td><td>100%</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Student requirements</td><td>The student should have a computer and internet connection.</td></tr>
<tr><td>Course policies</td><td>A- Attendance policies:
• Attendance is required. Class attendance will be taken every class and the university polices will be enforced in this regard.
B- Absences from exams and not submitting assignments on time:
• A makeup exam can be arranged for students with acceptable absence causes.
• Assignments submitted late, but before announcing or discussing the solution can be accepted with 25% penalty.
• The project report must be handed in in time.
C- Health and safety procedures:
• All health and safety procedures of the university and the school should be followed.
D- Honesty policy regarding cheating, plagiarism, misbehavior:
• Open-book exams
• All submitted work must be of the submitting student.
• Other text or code must be properly quoted with clear source specification.
• Cheating will not be tolerated.
E- Available university services that support achievement in the course:
• Microsoft Teams team and Moodle course page
• AI Lab for practicing the practical aspects and solving the programming assignments.
• Program announcements Facebook group</td></tr>
<tr><td>Additional information</td><td>None</td></tr>
</table>

Appendix

Learning Outcomes for the BSc in Intelligent Systems Engineering

Students who complete the BSc in Intelligent Systems Engineering (CprE) will be able to:

1. Identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
2. Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.
3. Communicate effectively with a range of audiences.
4. Recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.
5. Function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.
6. Develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.
7. Acquire and apply new knowledge as needed, using appropriate learning strategies.