<table>
<tr><td>Course Title</td><td colspan="5">Mechatronics</td></tr>
<tr><td>Course Code</td><td colspan="5">ME413</td></tr>
<tr><td>Course Type</td><td colspan="5">Compulsory</td></tr>
<tr><td>Level</td><td colspan="5">BSc (Level 1)</td></tr>
<tr><td>Year / Semester</td><td colspan="5">4th / Spring</td></tr>
<tr><td>Teacher’s Name</td><td colspan="5">Dr. Andreas Tjirkallis</td></tr>
<tr><td>ECTS</td><td>6</td><td>Lectures / week</td><td>3</td><td>Laboratories/week</td><td>1</td></tr>
<tr><td>Course Purpose</td><td colspan="5">The aim of the course is to provide students with the knowledge of the principles in developing an interdisciplinary understanding and integrated approach to mechatronic systems. It enables the development of the required skills in combining mechanical systems, signal processing, electrical and electronic systems with modern controls and microprocessors, in the analysis, design, synthesis and selection of systems and is vital to industrial automation and robotics.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="5">On completion of the course, students will be able to:
1. Explain the architecture of mechatronic systems
2. Analyze and synthesize modern electromechanical systems
3. Translate simple problems involving mechatronic control systems into a form solvable using mathematical models
4. Exploit the underlying similarities between the different physical fields (mechanical, electrical, hydraulic and thermal) to create abstractions for analysis, synthesis and design of mechatronic systems.
5. Design simple controllers to regulate the behaviour (e.g. motor speed) of a mechatronic system
6. Solve problems regarding the analysis, design and operation of mechatronic systems using Matlab, Octave and LabView.
7. Design, develop, program and test a simple mechatronic control system, working effectively in a small group, by implementing a microcontroller (i.e. Arduino) and/or a simple general-purpose computer (i.e. Raspberry Pi), data acquisition and an Integrated Development Environment (LabView, Arduino, Python)</td></tr>
<tr><td>Prerequisites</td><td colspan="2">ME211, ME323, ME327</td><td colspan="2">Corequisites</td><td>None</td></tr>
<tr><td>Course Content</td><td colspan="5">Mechatronic Systems: Importance of mechatronics in contemporary engineering design. Evolution of Mechatronics. Primary elements and functions of a mechatronic system.</td></tr>
</table>

	Modelling in Mechatronics Design: Modelling as part of a Design Process. Modelling of Systems and Signals. Signal conditioning and information processing. Classification of process elements. Fundamental equations of process elements. Connection of process elements. **Dynamic Models and Analogies:** Model Types. State-Space Representation. Model Linearization. Transfer-Function Models. Mechanical elements, Electrical elements, Thermal elements, Fluid elements. **Performance Specification and Analysis:** Parameters for performance specification. Time and Frequency Domain Specifications. Instrument Ratings. Bandwidth design of a Mechatronic System. **Sensors and Transducers:** Signal types and measuring amplifiers. Motion, Variable-Inductance, Self-Induction, Permanent-Magnet, Eddy Current and Variable-Capacitance transducers. Digital transducers. Piezoelectric Sensors, Torque sensors, Strain Gages, Gyroscopic sensors, Optical and Ultrasonic Sensors and Lasers, Thermo-fluid sensors. **Actuators:** Electromechanical actuator drives. Stepper Motors. Continuous Drive Actuators. Linear, Hydraulic and Pneumatic Actuators. Hydraulic and Fluidic Control Systems. **Microprocessors and PLCs:** Microprocessors, Arduino IDE, Python, LabView Programming Languages. Programmable Logic Controllers (PLCs) Data Acquisition and Control. **Control of Mechatronic Systems:** Control System Architectures and Performance. Instrumentation and Design. Frequency Domain Analysis. Controller Design and Tuning. Computer Implementation. **Design of Mechatronic Systems:** Intelligent Mechatronic Devices (IMS). Sensory perception. Mechatronic Systems Case Studies. Technology needs, machine features and hardware development. **Mechatronic Systems and Robotics:** Mechatronic Systems and robotization, taking into consideration process requirements, cost and economic realities, time constrains of implementation and production, as well as human factors.
Teaching Methodology	Students are taught the course through lectures (3 hours per week) in classrooms or lectures theatres, by means of traditional tools and using computer demonstration. Auditory exercises, where examples regarding matter represented at the lectures, are solved and further, questions related to particular open-ended topic issues are compiled by the students and answered, during the lecture or assigned as homework. Topic notes are compiled by students, during the lecture which serve to cover the main issues under consideration and can also be downloaded from the e-learning platform or the lecturer's webpage. Students are also advised to use the subject's textbook or reference books for further reading and practice in

	solving related exercises. Tutorial problems are also submitted as homework and these are solved during lectures or privately during lecturer's office hours. Furthermore design projects are assigned to the students, where literature search is encouraged to identify a specific problem related to some issue, gather relevant scientific information about how others have addressed the problem, analyse and implement the design, and report this information orally in a project demonstration session as well as in a written report. Students are prepared for final exam, by revision on the matter taught, problem solving and concept testing and are also trained to be able to deal with time constraints and revision timetable.
Bibliography	**(a) <u>Textbooks:</u>** • D.G. Alciatore and M.B. Histand, ***Introduction to Mechatronics and Measurement Systems***, McGraw-Hill, 4th Edition, 2012. • C.W. De Silva, ***Mechatronics, an Integrated Approach***, CRC Press, 2005. **(b) <u>References:</u>** • W. Bolton, ***Mechatronic:*** Electronic control systems in Mechanical and Electrical Engineering, Pearson, 6th Edition, 2015 • R. Isermann, ***Mechatronic Systems***, Springer, 2005. • R.H. Bishop, ***The Mechatronics Handbook,*** CRC Press, 2nd Edition, 2008. • M. S chwartz and O. Manickum, ***Programming Arduino with LabView,*** Packt Publishing, 2015.
Assessment	The final assessment of the students is formative and summative and is assured to comply with the subject's expected learning outcomes and the quality of the course. The Students are assessed via continuous assessment throughout the duration of the Semester, as well as with a laboratory and project work, which forms the Coursework grade and the final written exam. The coursework and the final exam grades are weighted 60% and 40%, respectively, and compose the final grade of the course. Assignments range from simple problems to work out, to LabView and Matlab assignments that require concept understanding as well as problem-solving skills, encouraging and reinforcing learning. An important part of the continuous assessment is the design, development, programming and presentation of a mechatronic system project. The assessment weight, date and time of each type of continuous assessment is being set at the beginning of the semester via the course outline. An indicative weighted continuous assessment of the course is shown below: • Assignments 10%

	• Mid-Term written exams 15% • Laboratory Work 10% • Project Work 25% • Final Exam 40% Students are prepared for final exam, by revision on the matter taught, problem solving and concept testing and are also trained to be able to deal with time constrains and revision timetable.
Language	English