COURSE DESCRIPTION CARD - SYLLABUS

Course name

Programming of control systems in mechatronics

Course

Field of study

Mechatronics

Year/Semester

2/3

Area of study (specialization)

Design and control of mechatronic devices

Profile of study

general academic

Level of study

Second-cycle studies

Course offered in

Polish

Form of study

full-time

Requirements

elective

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
	30	
Tutorials	Projects/seminars	

Number of credit points

2

Lecturers

Responsible for the course/lecturer:

Msc Tymoteusz Lindner

email: tymoteusz.lindner@put.poznan.pl

Faculty of Mechanical Engineering

Piotrowo 3, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

Basic knowledge of mathematics, computer science, and programming languages. Knowledge of computer, Windows, and Linux operating systems and programming in C++, C#, and Python, building algorithms using elements of C++, C#, and Python to control mechatronic devices. Awareness of the need to expand knowledge and skills. Ability to follow the rules of the laboratory classes.

Course objective

Acquiring the knowledge of the basics of object-oriented programming, and acquiring the ability to use classes and structures. Learning the basics of controlling mobile robots and robot arms. The ability to control robots in a simulation environment. Learning and designing software for robot control. Autonomous robot control.

Course-related learning outcomes

Knowledge

Has extended knowledge of robot control, including mobile robots and robotic arms.

Has basic knowledge of programming autonomous control systems for mobile robots.

Has extensive knowledge of programming in C++, C#, and Python. Has basic knowledge of the construction, operation, and programming of control systems.

Has extended knowledge of mechatronics in the field of analysis and design of complex mechatronic systems, systems theory, and the application of modeling and simulation in mechatronic design.

Has extended knowledge of computer science with the knowledge of real-time systems operation, programming with the use of algorithms for signal processing and control, the basics of image processing and analysis, and preparation of documentation.

Skills

Ability to obtain information on mechatronics from the Internet, libraries, and from other sources. In particular, he is able to correctly indicate the sources of the necessary information.

Ability to write application programs in C++, C#, and Python. Can write and use programs for design, analysis, simulation, and control.

Has the ability to use IT tools in acquiring and integrating information, designing and controlling mechatronic devices.

Social competences

Understanding the requirement of learning by whole life; ability to inspire and organize the learning process of other people.

Is aware of the role of autonomous robots and their importance for the development of society and the environment.

Ability to think and act in a creative and enterprising way.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Credit based on reports from laboratory exercises and written assignments consisting of programming tasks.

Grading scale:

<51%-60%> points - 3.0,

(60%-70%> points - 3.5,

(70%-80%> points - 4.0,

(80%-90%> points - 4.5,

(90%-100%> points - 5.0.

Rewarding the practical knowledge gained during the previous laboratory exercises.

Practical check of robot programming skills.

Earning additional points for activity in the classroom, especially for:

• the ability to work in a team that practically performs a specific task in the laboratory,

• performing additional tasks,

• aesthetic care of completed projects.

Programme content

• Programming in C++, C#, and Python.

• Installation and configuration of the Ubuntu operating system.

• Installation and configuration of the ROS system.

• Installation and configuration of the simulation environment.

• Basics of the Linux operating system, Embedded Linux and ROS.

• Design, construction, and programming of the control system for mobile robots and robotic arms in the ROS environment.

• Simulation of control systems in simulation environments.

• Development of software for the operation of selected elements of the control system.

• Autonomous control of mobile robots..

Teaching methods

Individual practical exercises, performing experiments, solving problems, discussion, teamwork

Bibliography

Basic

1. ROS Documentation

2. ROS Robotics Projects, Lentin Joseph

3. Mastering ROS for Robotics Programming, Lentin Joseph, Jonathan Cacace

Additional

1. Learning Python, Mark Lutz

2. C Primer Plus, Stephen Prata

Breakdown of average student's workload

	Hours	ECTS
Total workload	50	2,0
Classes requiring direct contact with the teacher	30	1,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	20	1,0

[1] delete or add other activities as appropriate