COURSE DESCRIPTION CARD - SYLLABUS

Course name
Information engineering [S1AiR1E>Inf1]

Course

Field of study Automatic Control and Robotics	Year/Semester 1/1
Area of study (specialization) –	Profile of study general academic
Level of study first-cycle	Course offered in English
Form of study full-time	Requirements compulsory

Number of hours

Lecture 60	Laboratory classes 30	Other 0
Tutorials 0	Projects/seminars 0	

Number of credit points

8,00

Coordinators

dr inż. Piotr Kaczmarek
piotr.kaczmarek@put.poznan.pl

Lecturers

Prerequisites

required knowledge in the fields of high school (computer skills, mathematics, and computer science)

Course objective

Educational Module Objective: 1. Familiarize with the methodology and principles of structured and object-oriented programming using the C++ programming language, including procedural and object-oriented programming. 2. Familiarize with dynamic data structures and their implementation in C++ and Python. Develop practical skills for the appropriate use of structures depending on requirements. 3. Ability to implement and adapt standard algorithms to solve various problems, as well as issues related to computational complexity and optimization. 4. Knowledge of contemporary issues concerning computer architecture.

Course-related learning outcomes

Knowledge:
Has advanced structured knowledge of selected algorithms and data structures as well as procedural and object-oriented programming methodologies and techniques [K1_W8 (P6S_WG)].
Has a structured knowledge of computer architectures, computer systems and networks and operating

systems including real-time operating systems [K1_W9 (P6S_WG)].
Has a basic knowledge of the handling and use of IT tools for the design, rapid prototyping, simulation and visualisation of automation and robotics systems and for recording the design of mechanical constructions [K1_W10 (P6S_WG)].
Skills:
Is able to obtain information from literature, databases and other sources also in a chosen foreign language [K1_U1 (P6S_UW)].
Is able to plan, prepare and simulate the operation of simple automation and robotics systems [K1_U10 (P6S_UW)].
Is able to develop a solution to a simple engineering task and implement, test and run it in a selected programming environment on a PC for selected operating systems [K1_U26 (P6S_UW)].
Social competences:
Is aware of the importance and understands the non-technical aspects and consequences of engineering activities, including their impact on the environment and the related responsibility for decisions; is ready to care for the achievements and traditions of the profession [K1_K2 (P6S_KR)].
The graduate is aware of the need for a professional approach to technical issues, meticulous familiarization with the documentation and environmental conditions in which the equipment and its components can operate. The graduate is ready to observe the rules of professional ethics and to demand it from others, to respect the diversity of opinions and cultures [K1_K5 (P6S_KR)].

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture: written exam covering the lecture material
Laboratory: assessment of practical skills in algorithms and data structures, object-oriented programming in C++, and the ability to use C++ libraries STL and SFML.
The grade for the first semester is based on the average of two quizzes, class participation, and homework assignments.
The laboratory grade for the second semester is based on the average of a quiz, class participation, homework assignments, and the final project evaluation.

Programme content

The subject covers 2 thematic blocks:
1. Introduction to programming in C++
 - basic data types
 - control flow
 - functions
 - introduction to algorithms
 - dynamic data structures
 - the STL library including containers, algorithms, regular expressions
 - introduction to object-oriented programming
2. Computer architecture
 - data storage
 - computer graphics
 - interfaces and computer networks

Course topics

The lecture covers:
1. Introduction to programming in C++
- basic data types
- flow control
- functions
- introduction to algorithms
- dynamic data structures
- STL library including containers, algorithms, regular expressions
- introduction to object-oriented programming
2. Computer architecture
- data storage

- computer graphics
- interfaces and computer networks

Laboratory classes in the first semester include:
- introduction to programming in C++
- algorithms, dynamic data structures in the STL library

Laboratory classes in the second semester include:
- introduction to object-oriented programming
- basics of computer graphics in the SFML library
- practical application of knowledge in preparing the final project

Teaching methods

1. Lecture: multimedia presentation, illustrated with examples given on the board, and with programs created during the classes.
2. Laboratory exercises: practical exercise on C++, supported by didactic materials placed on the e-learning platform

Bibliography

1. B. Eckel, Thinking In C++,
2. teaching materials available for laboratory classes and lectures
https://ekursy.put.poznan.pl
3. Brad Miller and David Ranum “Problem Solving with Algorithms and Data Structures using Python” Luther College 2018 (dostępna online)

Breakdown of average student's workload

	Hours	ECTS
Total workload	180	8,00
Classes requiring direct contact with the teacher	60	4,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	90	4,00