POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name

Object-oriented programming

Course

Field of study	Year/Semester
Automation and Robotics	1/1
Area of study (specialization)	Profile of study
Automation and robotics systems	general academic
Level of study	Course offered in
Second-cycle studies	polish
Form of study	Requirements
part-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
12	12	-
Tutorials	Projects/seminars	
-	-	

Number of credit points

4

Lecturers

Responsible for the course/lecturer:	Responsible for the course/lecturer:
Dr. Eng. Wojciech Kowalczyk	

e-mail: Wojciech.Kowalczyk@put.poznan.pl

phone: 61 6652043

Faculty of Control, Robotics and Electrical Engineering

Piotrowo 3, 60-965 Poznan

Prerequisites

Knowledge: The student starting this course should have basic knowledge of structured programming and be able to design and write a simple program.

Skills: He/she should understand the necessity of expanding his competences and be ready to cooperate within the team.

Social competences: In addition, in terms of social competences, the student must present attitudes such as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

The aim of the module is to develop students' knowledge of programming with object-oriented techniques. Students gain knowledge in this field during the lecture and practice skills in laboratories. As part of the classes, students develop teamwork skills at various stages of the project from analysis of assumptions to implementation and testing.

Course-related learning outcomes

Knowledge

1. He/she has an organized, theoretically founded, detailed knowledge of software analysis and design methods; - [K2_W7]

2. He/she has extended knowledge within selected areas of robotics; - [K2_W10]

3. He/she has theoretically founded detailed knowledge related to programming - [K2_W11]

Skills

1. He/she can make critical use of literature information, databases and other sources in Polish and in a foreign language; - [K2_U1]

2. He/she is able to analyze and interpret technical design documentation and use scientific literature related to a given problem; - [K2_U2]

3. He/she is able to integrate and program specialized robotic systems; - [K2_U12]

Social competences

1. He/she is aware of responsibility for own work and is ready to submit to the rules of teamwork and responsibility for jointly performed tasks, can set goals and define priorities leading to the implementation of the task - [K2_K3]

2. He/she is aware of the need for a professional approach to technical issues, scrupulous reading of the documentation and the environmental conditions in which the devices and their components may function - [K2_K4]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:
Formative assessment:

In the field of lectures:

- on the basis of answers to questions about the material discussed in previous lectures, In the field of laboratories:

- on the basis of an assessment of the current progress of work,

Summative assessment:

a) in the field of lectures, verification of the assumed learning outcomes is carried out by:

1. assessment of the knowledge and skills shown on the written test, during which the student answers 5 questions selected from 30 previously provided to students and one question requiring problem analysis. The maximum number of points in the exam is 30, in order to obtain a satisfactory grade, the student must obtain at least 15 points

2.discussion of the test results,

b) in the field of laboratories, verification of the assumed learning outcomes is carried out by:

3.assessment of the performance of works and skills related to their implementation,

4.assessment of teamwork skills,

5. evaluation and? Defense? reports on the implementation of work by the student (the report describes the work carried out in the field of analysis, design and implementation as well as tests),

Obtaining additional points for activity during classes, especially for:

1. discuss additional aspects of the issue,

2.effectiveness of applying the acquired knowledge while solving a given problem,

3.the ability to cooperate within a team that practically carries out detailed tasks,

4. identifying students' perceptual difficulties enabling ongoing improvement of the teaching process.

Programme content

Differences between structural and object-oriented programming. Characteristics of selected object-oriented programming languages in terms of code portability, memory management, speed.

The concept of class and object, procedural abstraction and data abstraction, compositional and generalizing abstraction.

Encapsulation, advantages of using it; protection of members by granting access rights to object members. Initialization of variables in the class. Constructors and destructors; constructor initialization list; copy constructor. Static fields and class methods - properties and applications. Constant objects and constant member functions. Object arrays. Operators and their reloading, ways of reloading operators.

Building a hierarchy of classes, the concept of inheritance, multi-generational inheritance, multiple inheritance. Polymorphism. The mechanism of virtuality. The concept of abstraction and the idea of an abstract class.

Review of selected design patterns and discussion of their applications.

Laboratory classes are carried out as part of nine two-hour meetings, during which students in groups of two carry out tasks consisting in the preparation of programs in which they use object-oriented programming techniques.

Teaching methods

1. lecture: multimedia presentation, presentation illustrated with examples given on the blackboard.

2. project activities: team work, workshops, discussion, performing experiments.

Bibliography

Basic

1. Siddhartha Rao, C++. Dla każdego, Helion

2. Adam Drozdek, C++. Algorytmy i struktury danych, Helion

Additional

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4
Classes requiring direct contact with the teacher	25	2
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	75	2

[1] delete or add other activities as appropriate