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

Informatics II

Course

Field of study

Electronics and Telecommunications

Area of study (specialization)

Level of study

First-cycle studies

Form of study

full-time

Year/Semester

2/4

Profile of study

general academic

Course offered in

Polish

Requirements

compulsory

Number of hours

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

Number of credit points

5

Lecturers

Responsible for the course/lecturer:

prof. dr hab. inż. Grzegorz Danilewicz,

grzegorz.danilewicz@put.poznan.pl

Responsible for the course/lecturer:

Prerequisites

Student has knowledge in the field of computer science and knows the syntax of programming languages as C, C ++, has the ability to learn independently, is aware of the need to expand their competences

Course objective

To teach students about the principles of object-oriented programming and the use of .NET libraries

Course-related learning outcomes

Knowledge

1. Student has a basic knowledge of development trends in the field of: high-level programming languages included in the .NET platform, organization of modern programming platforms and integration of programming languages.

2. Student knows high level programming language C#. (S)he knows the principles of designing computer programs with an extensive graphical interface, creating multithreaded software, cooperation with databases and the basic mechanisms related to programming network applications using the C# language.

Skills

1. Student can use the high-level programming languageC #; can design and program applications with an extensive graphic interface; knows how to create software implementing basic network protocols and cooperating with basic database suppliers; can design and create multithreaded software.

2. Student can solve typical engineering programming problems with the use of C# language.

Social competences

1. Student knows the limitations of his own knowledge and skills in the field of modern high-level programming languages, understands the need for further training in programming languages and platforms

2. Student is aware of the need for a professional approach to solving technical problems with the use of modern languages and programming platforms; student is aware of the responsibility taken for the software (s)he develops.

3. Student is aware of the dangers of badly designed and developed software, both for users and devices.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lectures. Knowledge is checked during two tests. Written tests concern the lectures content - one in the middle of the lectures, the other at the end of the lectures. Each of the tests must be passed with at least a satisfactory grade. A satisfactory grade is issued when the number of points for the test is greater than 50%. The final grade is the arithmetic mean of the test grades. For the test grades and the final grade, a grading scale from 2 (unsatisfactory - negative) to 5 (very good) is used. Rules for converting the final grade:

Average range : grade

0,00 - 2,99 : 2,0

3,00 - 3,24 : 3,0

3,24 - 3, 74 : 3,5

3,75 - 4,24 : 4,0

4,25 - 4,74 : 4,5

4,75 - 5,00 : 5,0

Laboratories. The following components are assessed: student's knowledge before the exercise, answers to questions during the exercises, written reports on the implementation of the exercises, written test at the end of the semester. For the component grades and the final grade, a grading scale from 2 (unsatisfactory - negative) to 5 (very good) is used.

Programme content

Introduction to programming in C# and the programming environment: .NET platform. Types, arrays and flow control. Classes and pillars of the object-oriented programming. Operators overloading. Inheritance and interfaces. Delegates and event handling. Handling of exceptions. Data structures and generic types. Multithreaded programming.

Teaching methods

Lecture with a whiteboard and/or projector, seminar lecture, experiment, case study, software writing.

Bibliography

Basic

1. Jesse Liberty "Programowanie C#", Helion 2005

2. https://docs.microsoft.com/en-gb/dotnet/csharp/

Additional

https://docs.microsoft.com/en-us/dotnet/core/

https://docs.microsoft.com/en-us/dotnet/architecture/

https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview

https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,0
Classes requiring direct contact with the teacher	70	3,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	55	2,0

[1] delete or add other activities as appropriate