



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Informatics II [N1EiT1>INF]

Course

Field of study

Electronics and Telecommunications

Year/Semester

3/5

Area of study (specialization)

–

Profile of study

general academic

Level of study

first-cycle

Course offered in

polish

Form of study

part-time

Requirements

compulsory

Number of hours

Lecture

30

Laboratory classes

0

Other (e.g. online)

0

Tutorials

35

Projects/seminars

0

Number of credit points

8,00

Coordinators

dr hab. inż. Mariusz Żal
mariusz.zal@put.poznan.pl

Lecturers

Prerequisites

The student taking the course should have a basic knowledge of computer design and an understanding of number systems. They should also be proficient in using high-level programming languages such as C and C++. Additionally, they should be able to gather information from literature, databases, and other sources in both Polish and English. Furthermore, they should possess the skills to integrate the acquired information, interpret it, draw logical conclusions, and provide justifications for their opinions.

Course objective

The purpose of the course is to familiarize the student with the principles of creating programs written in C#. As part of the course, the student will learn the principles of using control instructions and basic variable types. The student will also learn the principles of creating graphical applications and become familiar with programming models for database and network applications.

Course-related learning outcomes

Knowledge:

1. Knows the principles of computer program construction, has knowledge of computer science, and knows the syntax of the C# programming language.

2. Knows the utilization of computer memory, including reservations and references.

Skills:

1. Can proficiently use the high-level programming language C#. Can develop simple console and graphical programs.
2. Capable of writing basic programs for network communication and accessing databases.
3. Demonstrates the ability to engage in independent learning.
4. Can effectively communicate in both Polish and English within professional and other contexts.

Social competencies:

1. Possesses a sense of responsibility for the design of electronic and telecommunications systems and recognizes the potential hazards associated with their improper use. Understands the principles of information storage and access control in databases to ensure data security.
2. Recognizes the influence of telecommunications and ICT systems and networks on the development of the information society.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Wiedza zdobyta w ramach wykładu jest weryfikowana przez egzamin w formie pisemnej lub ustnej. W formie pisemnej studenci muszą udzielić odpowiedzi na 7-10 pytań (testowych i otwartych) różnie punktowanych. Istnieją trzy lub cztery grupy punktowej. Natomiast w przypadku egzaminu ustnego, student losuje jedno pytanie z każdej grupy punktowej. Dodatkowo, do każdego wylosowanego pytania student może otrzymać dodatkowe pytanie (związane z wylosowanym pytaniem). Ocena pytania (obejmująca zarówno odpowiedź na wylosowane pytanie, jak i pytanie dodatkowe) uwzględnia zakres odpowiedzi oraz głębokość zrozumienia zagadnienia. Przygotowuje się 50-60 pytań dla każdego egzaminu. Warunkiem zaliczenia egzaminu jest uzyskanie co najmniej 50% możliwych do zdobycia punktów.

Umiejętności nabyte w ramach ćwiczeń są weryfikowane na podstawie zadania realizowanego podczas ostatnich zajęć. Zadanie to jest podzielone na 5-6 podzadań różnie punktowanych. Podzadania stanowią całość, ale można je wykonywać niezależnie. Brak wykonania jednego podzadania nie wpływa na ocenę pozostałych podzadań. Próg zaliczenia wynosi 50% punktów.

Kryteria oceny egzaminu i zaliczania:

liczba punktów ocena

<=50 % 2,0

51% - 60% 3,0

61% - 70% 3,5

71% - 80% 4,0

81% - 90% 4,5

91% - 100% 5,0

Programme content

Lectures:

1. .NET programming basics: common runtime environment, common file system, intermediate language.
2. Basics of object-oriented programming in C#.
3. Exception handling.
4. Interfaces and collections.
5. Memory cleanup mechanisms.
6. Operator overloading.
7. Delegates and events.
8. Creating applications using Windows Forms.
9. Creating applications using WPF.
10. Programming simple network applications.
11. Programming simple database applications using ADO.NET libraries.

Exercises:

1. Conditional instructions, loops, input validation functions (Parse, TryParse, etc.).
2. Creating classes - adding properties and methods, overloaded constructors.
3. Creating classes - adding properties that are student-defined types.
4. Dynamic data structures: one-way and two-way lists, binary trees.

5. Text editor in the form of a windowed application that allows writing and reading files from disk.
6. GUI support, event-driven programming using Windows.Forms.
7. File operations.
8. Data modeling, representation in custom structures, relationships.

Teaching methods

1. Lecture:
 - a) Multimedia presentations illustrated by examples given on the blackboard.
 - b) Case analysis performed directly in runtime environments for programming languages; students participate actively by asking questions and suggesting results that can be obtained.
2. Exercises: Performing tasks given by the instructor (practical exercises carried out using runtime environments for C# programming languages) supplemented by multimedia presentations.

Bibliography

Basic

1. Troelsen A. , Japikse P., Język C# 6.0 i platforma .NET 4.6, [przekład WITKOM Witold Sikorski: Maciej Baranowski], PWN, 2017.

Additional

1. Stephen C. Perry, C# i .NET, Helion 2006.

Breakdown of average student's workload

	Hours	ECTS
Total workload	175	8,00
Classes requiring direct contact with the teacher	75	3,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	100	5,00