COURSE DESCRIPTION CARD - SYLLABUS

Course name
Information systems [S1ETI1>SI]

Course

Field of study
Education in Technology and Informatics

Year/Semester
2/3

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
30

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr hab. inż. Małgorzata Jankowska
malgorzata.jankowska@put.poznan.pl

Lecturers

Prerequisites

Basic knowledge of computer science. The ability to program fluently in C/C ++, algorithmization of tasks as well as logical and abstract thinking. Understanding the need to create IT systems in order to increase work efficiency, calculations, visualization of results and presentation of information collected in databases.

Course objective

The main aim of the course is to provide knowledge on advanced elements of the C/C ++ language, the knowledge of which enables the creation of proprietary, complex information systems. In addition, an introduction to the programming language Java.

Course-related learning outcomes

Knowledge:
w02 knows and understands advanced elements of the c / c ++ language used to create it applications, including object-oriented programming, operator overloading, exception handling, dynamic data structures, namespaces. k1_w08
w03 has the necessary knowledge to create window applications running on windows. k1_w08

Skills:
u02 has the ability to create computer programs using advanced elements of the c / c ++ programming language. k1_u02, k1_u04, k1_u11
u03 has the ability to obtain information from literature, databases and other available sources of

Social competences:
k01 is able to share the acquired it knowledge with others in an understandable way. k1_k01, k1_k03, k1_k05
k02 sees the importance of computerization of various areas of life in relation to man and society. k1_k01, k1_k03, k1_k05
k03 understands the need for continuous training in order to improve professional competences, acquire current knowledge in the field of generally understood computer science and programming languages (e.g. by reading it magazines, participating in courses and postgraduate studies). k1_k01, k1_k03, k1_k05
k04 demonstrate responsibility for the tasks entrusted to him, e.g. an independent programming project. k1_k01, k1_k03, k1_k05
k05 is able to work on a designated task independently and in a team taking various roles in it. k1_k01, k1_k03, k1_k05

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

W01 Written exam verifying a knowledge and a proper understanding of the field of study. The final grade depends on the percentage of points:
 3 50.1%-70.0%
 4 70.1%-90.0%
 5 od 90.1%

U01 Project tasks required in the form of computer programs and evaluated at the end of laboratory classes and/or during exams. The final grade depends on the percentage of points:
 3 50.1%-70.0%
 4 70.1%-90.0%
 5 od 90.1%

Programme content

The subjects of the module includes advanced C++ language elements used to create functions and also object oriented programming in C++. The next group of issues concerns the introduction to programming in Java. The curriculum content assumes a discussion of the basics of Java and elements of object-oriented programming in this language.

Course topics

Lectures:
1. Object oriented programming:
- classes, class objects,
- operator overloading,
- exception handling,
- Inheritance,
- polymorphism.
2. Introduction to Java programming language:
- similarities and differences between C++ and Java,
- the advantages of Java and the possibilities of using this language to create applications,
- object-oriented programming in Java.

Laboratories:
1. Advanced C ++ language elements used to create functions:
- static and dynamic arrays, one- and two-dimensional as arguments of a function,
- recursion,
- const modifier,

- pointers and references as a function result,
- function overload,
- function templates.
2. Object oriented programming:
- classes, class objects,
- operator overloading,
- exception handling,
- Inheritance,
- polymorphism.
3. Introduction to Java programming language:
- similarities and differences between C++ and Java,
- the advantages of Java and the possibilities of using this language to create applications,
- object-oriented programming in Java.

Teaching methods

Lectures:
Multimedia presentation illustrated with example programs written in the C/C++ language.

Laboratories:
Writing of computer programs in the C/C++ language.

Bibliography

Basic:
1. H. M. Deitel, P. J. Deitel, Arkana C++ Programowanie, Wydawnictwo RM, Warszawa 1998.
2. S. Prata, Szkoła Programowania. Język C++, Wydawnictwo Helion, Gliwice 2006.
3. A. Zalewski, Programowanie w językach C i C++ z wykorzystaniem pakietu Borland C++, Wydawnictwo Nakom, Poznań 1996.
4. J. Grębosz, Symfonia C++. Programowanie w języku C++ orientowane obiektowo, Tom 1,2,3, Oficyna Kallimach, Kraków 1999.
5. H. Schildt, Java. Kompendium programisty, Wydanie XII, Helion, Gliwice 2023.
6. K. Sierra, B. Bates, T. Gee, Java. Rusz głową! Przewodnik po praktycznym programowaniu w Javie, Wydanie III, Helion, Gliwice 2023.

Additional:
1. D. E. Knuth, Sztuka programowania,tom1. Algorytmy podstawowe, Wydawnictwa Naukowo-Techniczne, Warszawa 2002.
2. N. Wirth, Algorytmy + struktury danych = programy, Wydawnictwa Naukowo-Techniczne, Warszawa 2004.

Breakdown of average student's workload

	Hours	ECTS
Total workload	80	3,00
Classes requiring direct contact with the teacher	50	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	30	1,00