COURSE DESCRIPTION CARD - SYLLABUS

Course name
Programming Fundamentals - Delphi [N1Inf1>DELPHI]

Course

Field of study
Computing

Year/Semester
1/1

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
polish

Form of study
part-time

Requirements
elective

Number of hours

Lecture
12

Laboratory classes
20

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

4,00

Coordinators

prof. dr hab. Andrzej Marciniak
andrzej.marciniak@put.poznan.pl

Lecturers

prof. dr hab. Andrzej Marciniak
andrzej.marciniak@put.poznan.pl

Prerequisites

In accordance with the core curriculum of general education available on the website http://bip.men.gov.pl/men bip/akty_prawne/rozporzazenie_20081223_zal_4.pdf it is assumed that starting the subject, the student has basic knowledge and skills: - in mathematics - 4th educational stage, basic scope extended to include differential calculus (in the field of extended), - in computer science - 4th educational stage, basic scope. Moreover, in terms of social competences, the student must demonstrate such attitudes as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture and respect for others other people.

Course objective

The aim of the course is to familiarize the student with the basics of computer programming and to teach programming in Delphi (formerly Pascal). In particular this includes: - providing students with basic information about the development of programming languages and programming structural, principles of object-oriented programming and program construction text and window, - developing students' ability to algorithmize problems and their software, including the form functions and procedures, - teaching students how to use an integrated programming system fluently, - students' mastery of object-oriented programming techniques, including creation various program units and access to the data and codes contained in them, - teaching students how to use components and use them in programs, - students acquire the ability to programmatically protect codes against errors executing programs.

Course-related learning outcomes

Knowledge:
1. Has extended and deepened knowledge of mathematics useful for formulating and solving complex IT tasks regarding, among others, logic programming, formal specification and software verification [K1st_W1].
2. Has structured and theoretically based general knowledge of key issues computer science and detailed knowledge of selected issues in this discipline [K1st_W4].
3. Knows the basic techniques, methods and tools used in the process of solving tasks IT, mainly of an engineering nature {K1st_W7].

Skills:
1. Is able to use appropriately selected methods when formulating and solving IT tasks, including analytical, simulation or experimental methods [K1st_U4].
2. Has the ability to formulate algorithms and implement them using at least one from popular tools [K1st_U11].
3. Is able to organize, cooperate and work in a group, taking on various roles in it and is able to appropriately determine priorities for the implementation of the task specified by yourself or others [K1st_U18].

Social competence:
Is aware of the importance of knowledge in solving engineering problems and knows examples and understands the causes of malfunctioning IT systems.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:
- based on the answers provided regarding the implementation of tasks during the exercises.
Summative assessment
Checking the assumed learning outcomes is carried out by:
- assessment of skills related to the implementation of laboratory tasks (during the semester, each student has four programs to be written with gradually increasing difficulty),
- assessment of knowledge and skills demonstrated during the written final test (after the last one lecture) of a problematic and practical nature (the colloquium consists of 6 topics/tasks with different scales of difficulty and the resulting different scores; a maximum of 20 points can be obtained; on a grade of 3.0 requires 11 points).

Programme content

The course program covers the following topics:
- basic concepts related to programming (programming, algorithm, program, language programming, machine-oriented language, instruction, higher-order language, universal language, specialized language),
- overview of programming languages (Ada, Algol, assemblers, Basic, C, C++, Cobol, Fortran, Java, Lisp, Logo,

Pascal, PHP, PL/1, Prolog, Python, Ruby),
- action networks (flowcharts) and symbols used in them,
- programming maxims (based on D. Van Tassel's book "Programming Practice"),
- general principles of object-oriented programming (heredity, hermeticity
and polymorphism),
- general characteristics of the Embarcadero Delphi XE10 package (will be presented in the following years next versions along with their purchase for laboratories),
- basic concepts related to constructing programs in an integrated environment
Delphi programming (project, form, component, property, event),
- using the integrated Delphi programming environment,
- review of the Delphi language structure (program, module, library, functions and procedures, classes and objects,
data types, variables, instructions),
- program and module structure,
- basic elements of the language (basic symbols, keywords and language directives,
identifiers, numbers, strings including Unicode strings, logical literals, comments
and separators),
- data types and their description (defining types, simple types, string types, structured types describing objects,
indicators, type compatibility),
- variables (variable declarations, simple, indexed, record, object, dynamic variables,
procedural, variant, with initial value, variable overlap, constant and variable literals),
- expressions (types of operators and their priority, expression syntax, constant expression),
- instructions (simple, structured, assembly language),
- functions and procedures (definitions, parameter types, overloading, calling, conventions
calling),
- object processing (constructors and destructors, static, virtual, dynamic methods
and abstract, messaging, properties),
- dynamically linked libraries (creating libraries, static and dynamic function downloads
and procedures from libraries),
- packages,
- file processing,
- message windows.
During laboratory classes, students become familiar with the integrated environment
Delphi programming, they write programs using the learned elements of the language.

Teaching methods

1. Lecture - multimedia presentation (each lecture) and presentation of writing and performance of selected ones
programs directly in the integrated Delphi programming environment.
2. Laboratory classes - practical exercises on the elements of the Delphi language, writing programs windows in this language.

Bibliography

Basic
A. Marciniak, Język programowania Delphi, Wydawnictwo NAKOM, Poznań

Additional
1. X. Pacheco, S. Teixeira, Delphi 7 - vademecum profesjonalisty, tom 1, Wydawnictwo HELION, Gliwice.
2. M. Cantu, Delphi 7 - praktyka programowania, tom 1 i 2, Wydawnictwo MIKOM, Warszawa.
3. A. Marciniak, Borlan Delphi 5 Professional - Object Pascal, Wydawnictwo NAKOM, Poznań.

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	32	1,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	68	2,50