COURSE DESCRIPTION CARD - SYLLABUS

Course name
Introduction to Algorithmics [S1Inf1>WdA]

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 full-time	Requirements elective

Number of hours

Lecture 24	Laboratory classes 30	Other (e.g. online) 0
Tutorials 0	Projects/seminars 0	

Number of credit points

4,00

Coordinators	Lecturers
mgr inż. Artur Laskowski artur.laskowski@put.poznan.pl	mgr inż. Artur Laskowski artur.laskowski@put.poznan.pl

Prerequisites

In accordance with the core curriculum of general education available at: http://bip.men.gov.pl/men bip/akty_prawne/rozporzenia_20081223_zal_4.pdf it is assumed that when starting the subject, the student has basic knowledge: - in mathematics: educational stage IV, basic scope extended by differential calculus (extended scope); - in computer science: 4th educational stage, scope basic. In accordance with the core curriculum of general education available at: http://bip.men.gov.pl/men bip/akty_prawne/rozporzenia_20081223_zal_4.pdf it is assumed that when starting the subject, the student has basic skills: - in mathematics: educational stage IV, basic scope extended by differential calculus (extended scope); • in computer science: educational stage IV, scope basic. In terms of social competences, the student must demonstrate attitudes such as honesty, responsibility, perseverance, ambition and cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

1. Providing students with basic programming knowledge for effective implementation algorithms in C++ and programming techniques that improve the implementation process algorithms and optimizing their performance. 2. Providing students with knowledge of algorithmics in the field of basic numerical algorithms, binary search, number sorting, greedy algorithms and dynamic programming and mounds. 3. Providing students with knowledge of computational complexity in terms of determining and comparing complexity of algorithms operating in polynomial time. 4. Developing students' skills in programming implementation of the learned algorithms and data structures. 5. Developing students' skills in selecting the appropriate algorithm and data structure problem being solved and an assessment of the computational and memory complexity of their implementation. 6. Developing students' team programming skills in accordance with the ACM ICPC formula. 7. Developing students' skills in testing implemented algorithms and their evaluation. 8. Preparing students to take part in team programming competitions.

Course-related learning outcomes

Knowledge:
Has extended and deepened knowledge of mathematics useful for formulating and solving
complex IT tasks involving analysis and formal proof of correctness and
computational complexity of algorithms (K1st_W1)
Has structured, theoretically based general knowledge of algorithms and complexity
computational. (K1st_W4)
Knows the basic methods, techniques and tools used to solve simple tasks
IT in the field of analyzing the computational complexity of algorithms and problems. (K1st_W7)

Skills:
Is able to use analytical methods to formulate and solve IT tasks
experimental in order to select appropriate algorithms and data structures. (K1st_U4)
Has the ability to formulate algorithms and program them using C++. (K1st_U11)
Is able to cooperate and work in a group, taking on various roles during teamwork
solving a set of problems. (K1st_U18)

Social competence:
Understands based on the history of C++ standards ('11, '14, '17) that in computer science knowledge and skills
they become obsolete very quickly. (K1st_K1)
Is aware of the importance of knowledge in solving problems that require designing new ones
algorithms and knows examples and understands the causes of incorrectly implemented algorithms
led to serious financial losses. (K1st_K2)

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:
a) in the field of lectures, verification of the assumed learning outcomes is carried out by:
- monitoring and rewarding students' own work involving independent solving
algorithmic tasks
- rewarding students' activity during lectures
b) - monitoring the student's progress in solving algorithmic problems during classes
laboratory and outside them, which are assessed by an automatic checking system
Summary rating:
a) in the field of lectures, verification of the assumed learning outcomes is carried out by:
- colloquium requiring solving 4-10 practical problems in the field of algorithms,
- tasks are scored on a scale of 0-5 points (maximum number of points depends on the grade
difficulty), in increments of 0.5 points,
- points for tests are converted into percentages by dividing by the maximum number of points
to be obtained and counted with a weight of 0.3 in the final grade of the subject
b) in the field of laboratories, verification of the assumed learning outcomes is carried out by:
- analysis of the results of tasks completed during the semester - approximately 30 compulsory tasks,
including:

3.0 you have to solve 50% of the tasks, for 3.5 60% of the tasks, etc. - the grade is included in the final grade of the course
with a weight of 0.7
Additional assessment elements:
- for solving additional tasks selected by the instructor, the student can receive up to 1.5%
extra points for the task

Programme content

Program content included in this subject in addition to presenting algorithmic content
required of a graduate of first-cycle studies contain a wide range of topics that he has
prepare students to take part in algorithmic team programming competitions
compliant with the ACM ICPC formula. It is assumed that this subject has been chosen as an elective
will be mainly by students interested in this type of professions. Therefore, many
the aspects covered are discussed at an intermediate level and focused on them
practical use in solving problems arising, among others, on
algorithmic competitions.
The lectures begin with a presentation of the work system during the semester, which will be based on
automatic screening system, which will likely be new for many students
science. Therefore, a detailed presentation of the system and answers to the most common questions are provided
questions asked. Then, the concept of team programming based on
the ACM ICPC formula along with detailed rules and a discussion of the social competences that
should be worked out to increase the chance of achieving success in them. The next topic is issues
introducing algorithmics by discussing basic terms in the field of algorithmics,
such as problem and algorithm, data and data operations, instance, concept of type. She is moved
the subject of algorithm correctness, its definition and verification. These issues are illustrated
simple algorithms such as binary search and its application to string analysis
sorted, sequence of partial sums and differences, quick exponentiation and determination of Fibonacci numbers in
logarithmic time, calculating the values of polynomials, finding prime numbers and the greatest
common divisor. Based on the known algorithms, computational complexity is discussed
problems and the time and memory complexity of algorithms along with methods of determining it,
comparing and writing in O(), () and () notation. The problem of complexity is raised in
worst and best case and medium complexity. Then the algorithms are presented
sorting starting from the simplest ones, operating with quadratic complexity, such as sorting
bubble, by selection and insertion, by faster QuickSort sorting, by merging and Shell, by
sorting in linear time using bucket, positional and counting algorithms.

Teaching methods

1. Lecture: multimedia presentation illustrated with examples given on the board, solving
tasks.
2. Laboratory exercises: solving tasks, practical exercises, discussion, programming
team.

Bibliography

Basic:
1. Wprowadzenie do algorytmów, T.H. Cormen, Ch.E.Leiserson, R.L. Rivest, C. Stein, PWN, 2012
2. Algorytmika praktyczna nie tylko dla mistrzów, P. Stańczyk, PWN, 2009
3. Kombinatoryka dla programistów, W. Lipski, WNT, 2007
4. Tablice matematyczne, W. Mizerski, Adamantan, 2008
5. Symfonia C++ Standard, J. Grębosz, Edition 2000, 2008

Additional:
1. The CRC Concise Encyclopedia of Mathematics, E. W. Weisstein, CRC Press, 1998
2. Pasja C++, J. Grębosz, Edition 2000, 2010

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	54	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	46	2,00