Course description

Department/Unit /	KIV / ZEP			**Academic Year**	2023/2024
Title	Basics of effective programming			**Type of completion**	Exam
Accredited/Credits	Yes, 3 Cred.			**Type of completion**	Written
Number of hours	Lecture 2 [Hours/Week]				
Occ/max	Status A	Status B	Status C	**Course credit prior to**	YES
Summer semester	0 / -	0 / -	0 / -	**Counted into average**	YES
Winter semester	0 / -	0 / -	0 / -	**Min. (B+C) students**	10
Timetable	Yes			**Repeated registration**	NO
Language of instruction	Czech			**Semester taught**	Winter, Summer
Optional course	Yes			**Internship duration**	0
Evaluation scale	1|2|3|4			**Ev. sc. – cred.**	S|N
No. of hours of on-premise					
Auto acc. of credit	No				
Periodicity	K				
Substituted course	None				
Preclusive courses	KIV/ZEP-E				
Prerequisite courses	N/A				
Informally recommended courses	KIV/PPA1				
Courses depending on this Course	N/A				

Course objectives:

Students should master the basic principles used in the design of effective and robust algorithms. The course completes the set of the following existing courses of bachelor study: KIV/PPA1, KIV/PPA2, KIV/PT, KIV/PRO.

Requirements on student

Semester:

Four assignments including both theoretical exercises (e.g., design an effective algorithm that decides if there is no duplicity in the input array of integers) and practical (e.g., create a program that calculates the square root of two with 100 digits accuracy) are released during the semester. Each assignment is evaluated by at least 20 points. The student must collect at least 50 points to pass the course.

Exam:

Written test in which the student designs a method (algorithm) to solve a more complex problem. The test is evaluated by 20 points and these points are summed with points from the semester. Hence, the student can collect 100 points all in all. Marks are as follow: 1 = 86 - 100, 2 = 71 - 85, 3 = 56 - 70, 4 = < 55 points

Content

- The evaluation of algorithms, time and memory complexity, the robustness of algorithms.
- The accuracy of numerical calculations, Horner scheme, singular cases.
- Common redundant, time-consuming operations in the code and their elimination.
- Memory management. Memory allocation, deallocation, garbage collector, memory leaks. Too large data sets (they do not fit the memory).
- Matrix and vector operations (Strassen formula, dot and cross product). Linear and non-linear systems of equations (including overspecified systems).
- Cache in the current computers and its efficient use (bricking technique)

- Introduction to parallel and GPU programming. Space division, median.
- Recursion and its elimination. Binary vs. interpolation search.
- Reduction of problem space dimension (space-filling curves, PCA). Data sampling, Soboly sequences.
- Graph representations, Dijkstra and Floyd-Warshall algorithms. Graph matching - Hungarian marriage. State machine, pruning.
- Practical importance of compression, Freeman code. Checksums (LUHN, CRC, Adler32, MD5).

Fields of study

Guarantors and lecturers

- **Guarantors:** Doc. Ing. Josef Kohout, Ph.D. (100%)
- **Lecturer:** Doc. Ing. Josef Kohout, Ph.D. (100%)

Literature

- **Recommended:** Sedgewick, Robert. *Algorithms in Java. pt. 5, Graph algorithms*. 3rd ed. Boston : Addison-Wesley, 2004. ISBN 0-201-36121-3.
- **Recommended:** Wróblewski, Piotr. *Algoritmy : datové struktury a programovací techniky*. Vyd. 1. Brno : Computer Press, 2004. ISBN 80-251-0343-9.
- **Recommended:** McConnell, Jeffrey. *Analysis of algorithms: an active learning approach*. Jones & Bartlett Publishers, 2007. ISBN 978-0763707828.
- **Recommended:** McConnell, Steve. *Dokonalý kód : umění programování a techniky tvorby software*. Vyd. 1. Brno : Computer Press, 2005. ISBN 80-251-0849-X.

Time requirements

All forms of study

Activities	Time requirements for activity [h]
Individual project (40)	40
Preparation for an examination (30-60)	30
Contact hours	26
Total:	**96**

assessment methods

Knowledge - knowledge achieved by taking this course are verified by the following means:

Written exam

Seminar work

prerequisite

Knowledge - students are expected to possess the following knowledge before the course commences to finish it successfully:

Students are supposed to be interested in the topic and have the fundamental programming knowledge.

teaching methods

Knowledge - the following training methods are used to achieve the required knowledge:

Interactive lecture

Individual study

One-to-One tutorial

Seminar classes

learning outcomes

Knowledge - knowledge resulting from the course:

Upon completion of the course, students should have knowledge about common problems of the current software such as: ineffective use of computational power, not necessarily large memory requirements or unstable calculation; should master the basic principles of designing the code to avoid these problems and gain the experience in the design of various algorithms (some of those are from the ACM Contest).

Course is included in study programmes:

Study Programme	Type of	Form of	Branch	Stage	St. plan v.	Year	Block	Status	R.year	R.
Computer Science and Engineering	Bachelor	Full-time	Information Technologies	1	2018	2023	Doporučené výběrové předměty	C	1	LS