COURSE DESCRIPTION

1. Course title: THEORY OF COMPUTER SCIENCE
2. Course code: PI
4. Level of studies: 1st cycle of higher education
5. Mode of studies: intramural studies
6. Field of study: INFORMATICS (RAU)
7. Profile of studies: academic
8. Programme: ALL
9. Semester: 1, 2
10. Faculty teaching the course: Instytut Informatyki
11. Course instructor: dr inż. Alina Momot
12. Course classification: common subject
13. Course status: compulsory
14. Language of instruction: English
15. Pre-requisite qualifications: none
16. Course objectives:
The aim of the lecture is to delivery to students the information in the range of the basic notions of computer science. The aim of the classes and laboratory is to purchase by the students the skill in the range of creating the algorithms, low-level programming, understanding of works of microprocessors and introduction with the basic structures of the data.
The basic issues presented in the course are:
- the concept of the algorithm
- semaphores, classic synchronization problems (producer-consumer, problem of five philosophers)
- machine data representation and implementation of arithmetic operations
- organization of the central processor unit
- assembler language
- basic information about computer networks
- an overview of data access methods
17. Description of learning outcomes:

<table>
<thead>
<tr>
<th>Nr</th>
<th>Learning outcomes description</th>
<th>Method of assessment</th>
<th>Teaching methods</th>
<th>Learning outcomes reference code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A student knows formal methods of algorithms notation</td>
<td>Written test</td>
<td>Lecture, class, laboratory</td>
<td>K1A_W09</td>
</tr>
</tbody>
</table>

1 należy wskazać ok. 5 – 8 efektów kształcenia
2. A student has theoretically founded knowledge in the field of formal grammars. He can simulate simple algorithms using the Turing machine.

3. A student knows the basic construction elements of the computer and can discuss the role of each of them. He knows how the computer works according to the von Neumann model. He can explain the role of the computer control system, he can define simple instructions for the given lecture computer architecture (so-called W machine).

4. A student can write simple programs in the assembler language (W machine).

5. A student is able to define ways to protect indivisible resources and explain what is the management of resources and deadlock.

6. A student can use a queuing model to model a single computer and a computer network and analyse this model.

18. Teaching modes and hours

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester 1</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Semester 2</td>
<td>15</td>
<td>15</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

19. Syllabus description:

**Lecture:**
Algorithms. Concept and examples of algorithms, notation, serial and parallel implementation of algorithms. Complexity, time and memory complexity and relationships between them, examples of algorithm complexity evaluation for serial and parallel sorting algorithms.
The operating system and its tasks. The role and functions of the operating system. Synchronization and communication between processes. The concept of system resources.Sharable and unsharable resources. TAS procedure, semaphores. Producer-consumer problem. Resource management issues, deadlock protection.
Computer design elements. Registers, counters, encoders and decoders. The arithmetic and logic unit, its design and implementation. Buses and problems related to their design. Memories: types, design and implementation.
Interrupts, the concept, organization of the interrupt system. Interrupt priorities.
Information exchange between the central unit and the environment. Concepts of elementary I/O system. The exchange circuit as an intermediary between the central unit and external devices. Construction of an exemplary exchange circuit. Direct memory access (DMA), construction and operation.
Addressing modes.
Turing machine, description, Church-Turing thesis, a characteristic table, examples of programming algorithms for Turing machine.
Formal languages. Definition of grammar, examples of grammars, BNF notation, Łukasiewicz's algebra, the notion of translation and an example of translation algorithm of arithmetic expressions from infix notation to reverse Polish notation.
notation.

Data access methods. Lists, binary trees, network structures and their analysis in terms of search time. B-tree structures and task with optimal structure. Hashing functions and database structures based on them, collision problem.

Parallel computers. Notation of the algorithm in a canonical form, theorem on the implementation of the algorithm, serial and parallel implementation, definition of acceleration, model of data-flow computer and examples of algorithm implementation in such systems.

Computer networks and their statistical models. The concept of a computer network, media and their characteristics, network topologies, a layered model. Statistical computer model as a service station (M / M / 1) and its analysis.

Computer network model as M / M / 1 network. Open and closed networks. Methods for calculating load distribution in the network.

The latest trends in information technology. Information technology and genetics, quantum information systems.

Class

The subject of classes is closely related to lectures and is an extension and illustration of selected problems presented during the lectures. Some of the tasks are solved by the students themselves, when solving them or after solving, there is often a discussion that allows to consider alternative ways of solving the problem. The topics of table exercises include the following issues:

- Algorithms.
- Recursion and examples of recursive algorithms.
- Semaphores and resource management.
- Design of computer instructions
- Assembler programming
- Turing machine
- Formal grammars
- Data access methods
- Queue system M/M1 and networks

Laboratory

Laboratory exercises are conducted on the first semester of full-time studies. During the laboratory, students solve the tasks indicated by the teacher, using specially prepared software for this purpose. The following exercises are carried out as part of the laboratory:

1. Recursive algorithms
2. Synchronization and communication between processes
3. Instructions design for the machine W
4. Programming in the assembly language of the W machine
5. Input / output system
6. Interrupt system

20. Examination: yes (after the first semester)

21. Primary sources:

22. Secondary sources:
23. Total workload required to achieve learning outcomes

<table>
<thead>
<tr>
<th>Lp.</th>
<th>Teaching mode</th>
<th>Contact hours / Student workload hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lecture</td>
<td>45/45</td>
</tr>
<tr>
<td>2</td>
<td>Classes</td>
<td>45/45</td>
</tr>
<tr>
<td>3</td>
<td>Laboratory</td>
<td>30/30</td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>/</td>
</tr>
<tr>
<td>5</td>
<td>BA/MA Seminar</td>
<td>/</td>
</tr>
<tr>
<td>6</td>
<td>Other</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>Total number of hours</td>
<td>120/120</td>
</tr>
</tbody>
</table>

24. Total hours: 240

25. Number of ECTS credits: 8 (respectively in sem. 1 and 2: 6 + 2)

26. Number of ECTS credits allocated for contact hours: 4

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 2

26. Comments:

Approved:

(date, Instructor’s signature)  (date, the Director of the Faculty Unit signature)