



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Programming of network devices in Java and C#

Course

Field of study

Electronics and telecommunication

Area of study (specialization)

Computer networks and Internet technologies

Level of study

First-cycle studies

Form of study

full-time

Year/Semester

3/6

Profile of study

general academic

Course offered in

Polish

Requirements

elective

Number of hours

Lecture

15

Laboratory classes

15

Other (e.g. online)

Tutorials

Projects/seminars

Number of credit points

3

Lecturers

Responsible for the course/lecturer:

dr hab. inż. Mariusz Żal,
mariusz.zal@put.poznan.pl

Responsible for the course/lecturer:

Prerequisites

Students have a basic knowledge of computer networks and network protocols; Has a basic knowledge of C++, C#, and Java programming. Students are able to find information in literature, as well as other reference sources; is able to integrate and interpret obtained information, draws conclusions and justifies. Student understands a necessity to acquire a new knowledge and skills stemming from a chosen field of studies.

Course objective

To provide students with data exchanges mechanisms in compute networks and network sockets implementation in C# and Java. To create skills necessary to implement in Java or C# simple network applications based on TCP and UDP protocols. To prepare students programming database applications.

Course-related learning outcomes

Knowledge

1. Students knows the rules of designing computer programs, has a basic knowledge of computer science and knows the syntax of the C# and Java programming language.
2. Student has a systematic and well-founded knowledge of computer architectures.



3. Student has basic knowledge of standards, architecture, network protocols and operation of computer networks.
4. Student has a systematic and well-founded knowledge of network, transport, and application layer protocols.

Skills

1. Students are able to prepare network applications.
2. Students can select transport layer protocol and application layers protocol, depending on requirements.
3. Students can design and create multithreaded software .
4. Is able to use bibliography in English (books, scientific and technical journals, application notes, catalogs, instructions, recommendations etc.).

Social competences

1. Demonstrates responsibility for designed software. Is aware of the hazards they pose for individuals and communities if they are improperly designed.
2. A student is able to formulate opinions concerning challenges of contemporary networks application programming.
3. A student is aware of the impact of network application on the information society.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes are verified with a written or oral test. Test in written form consists 7-10 question (multichoice and open), which are scored in different ways (there are three or four groups of scores). Test in oral form contains 50-60 open questions divided into three or four score groups. Students draw one question from each group. In the oral form, for each question teacher can ask one additional question. Both, main and additional questions are scored, taking into account content range and understanding the issue. Minimum number of scores to pass the exam is equal to 50%.

Knowledge and skills gathered during laboratory classes are assessed by project realized on the last classes. The project contains 5 – 6 tasks which are scored in different ways. Each task can be realized independently. Minimum number of scores to pass the exam is equal to 50%.

The assessment levels (lecture and tutorials) are the following:

| Number of scores | mark |
|------------------|------|
| <=50 % | 2,0 |
| 51% - 60% | 3,0 |
| 61% - 70% | 3,5 |
| 71% - 80% | 4,0 |
| 81% - 90% | 4,5 |
| 91% - 100% | 5,0 |

Programme content



Lectures:

1. Network application architecture, layer 3 and 4 OSI RM
2. Addressing in layer 3 and 4, network sockets
3. Synchronous and asynchronous application, multithreading application
4. TCP layer application: FTP and TFTP
5. Applets
6. Servlets
7. Application layer mechanisms and protocols
8. Database application
9. Widows forms and database bindings
10. Multicast applications

Laboratory classes:

1. Simple synchronous network applications
2. Asynchronous network applications
3. Multithreading network applications
4. Database applications
5. Application layer: HTTP protocol
6. FTP I TFTP servers and clients
7. Java servlets

Teaching methods

1. Lectures:

- a) multimedia presentations illustrated with examples presented on the board.
- b) practical case study of selected events in database management systems or Java and C# programming languages runtime environment.

2. Laboratory classes : solving problems given by the teacher (practical case study with database management systems and Java and C# programming languages runtime environment) complemented with multimedia presentations.

Bibliography

Basic

1. Orłowski S., Grabek M., C# : tworzenie aplikacji sieciowych : gotowe projekty, Helion 2012
2. Hmrajani A., Bucka E., Java : tworzenie aplikacji sieciowych za pomocą Springa, Hibernate i Eclipse, Helion 2012
3. Reese G.. Java : aplikacje bazodanowe : najlepsze rozwiązania, Helion 2003.

Additional

1. Richard Blum, C# Network Programming, Sybex 2003
2. Anthony Potts, David H. Friedel Jr, Java programming language handbook, Coriolis Group Inc. 1996



Breakdown of average student's workload

| | Hours | ECTS |
|---|-------|------|
| Total workload | 75 | 3,0 |
| Classes requiring direct contact with the teacher | 31 | 2,0 |
| Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) ¹ | 44 | 1,0 |

¹ delete or add other activities as appropriate