Projekt "Perspektywy Współpraca Synergia Zarządzanie w Tarnowie" współfinansowany jest przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój. Projekt realizowany w ramach konkursu Narodowego Centrum Badań i Rozwoju z III Osi priorytetowej: Szkolnictwo wyzsze dla gospodarki i rozwoju; Działanie 3.5 Kompleksowe programy szkół wyzszych. Nr umowy o dofinansowanie projektu: POWR.03.05.00-00-Z087/17-00.

Module SYLLABUS

Organizational unit name	The Polytechnic Institute – Department of Computer Science		
Field of study	Computer science		
Module name	Distributed systems programming		
Module code	POWER.IP.4	**Erasmus code**	11.3
ECTS	3	**Module type**	Optional
Year of study	4	**Semester**	7

Form of classes	**Hours total**	**Form of assessment**
Project classes	30	Graded credit

Coordinator teacher	PhD Łukasz Mik		
Academic teacher	PhD Łukasz Mik		
Language of instruction	English		
Basic courses	No	**Open course / course at he another field of study**	No
Profile of education	Practical profile	**Study level**	First-cycle level

Prerequisites and additional requirements

Previous courses:
- Languages and programming techniques
- Object-oriented programming
- Architecture of computer systems
- Operating Systems
- Computer Networks

Additional requirements:
- Basic knowledge of C/C++ and Java languages
- Basic knowledge of client-server architectures

Learning outcomes for module

No.	**Student after module completion has the knowledge/knows how to/is able to Learning outcome code**	**Learning outcome type**	**Method of learning outcomes verification**	**Form of classes**
				Project
1.	Has the ability to self-education, among others to improve professional skills	Skills	Project	Y
2.	Is able to compare design solutions for distributed systems due to given operational and economic criteria.	Skills	Project, technical documentation	Y
3.	Can choose the right methodology for software development, use well-chosen programming environments, system modeling tools to design, implement and test software and information systems.	Skills	Project, technical documentation	Y

Projekt "Perspektywy Współpraca Synergia Zarządzanie w Tarnowie" współfinansowany jest przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój. Projekt realizowany w ramach konkursu Narodowego Centrum Badań i Rozwoju z III Osi priorytetowej: Szkolnictwo wyzsze dla gospodarki i rozwoju; Działanie 3.5 Kompleksowe programy szkół wyzszych. Nr umowy o dofinansowanie projektu: POWR.03.05.00-00-Z087/17-00.

4.	Can design, test and implement a designed application for a distributed system.	Skills	Project	Y
5.	Is able to develop documentation on the implementation of the engineering task and prepare a text containing a discussion of the results of this task	Skills	Technical documentation	Y
6.	Is aware of the responsibility for his own work and readiness to comply with the rules of working in a team and taking responsibility for the tasks he has carried out jointly.	Social competence	Project	Y

Didactic methods

Form of classes:
The project is implemented in two-person groups. All groups are obliged to present individual stages of the project during the semester.
Teaching methods:
The project carried out in stages, which are regularly verified by the teacher. Motivating to ask questions and solve them on the class forum - brainstorming.

Rules of assessment

In order to pass the project, the last week of the semester students should present a working system distributed within the local network in the computer lab. It is necessary to present point-to-point and point-to-multipoint tests. The grades are issued in accordance with the current regulations of studies at the State Higher Vocational School in Tarnow based on the design and complete technical documentation confirming its implementation.

Module content (brief)

In this module project classes are carried out with the following stages:

1. Choice of topic and specification of requirements for the final result of the project
2. Analysis of requirements and selection of appropriate tools
3. Implementation of a distributed system in one of the selected technologies: RPC, CORBA, RMI, JMS or Web Services.
4. Testing the call of remote services and the communication layer of the system
5. Design and implementation of a graphical user interface for a remote service client
6. Preparation of project documentation

Module content (comprehensive)

1. Choice of topic and specification of requirements for the final result of the project

   Choice of topic from several basic problems to be solved and final applied distributed data processing technology. The choices are tasks related to image processing, analysis of data from measurement systems, data processing from multiple files.

2. Analysis of requirements and selection of appropriate tools

   Preliminary activities related to the selected topic. Performing the analysis of requirements related to the size of data transferred between client-server points Based on the analysis results, the choice of one of the distributed system technologies: RPC, CORBA, RMI, JMS or Web Services. Studying the available literature of the subject in English.

3. Implementation of a distributed system in one of the selected technologies: RPC, CORBA, RMI, JMS or Web Services.
   The distributed system will consist of 2 main applications: server where the services and procedures will be available to remote objects and the client will send requests to server applications. During the course there is the opportunity to learn about different implementation methods:
   - RPC: server application, client application, interface description using the *rpcgen* tool, data coding in the XDR format, registration of the service under the selected port number using the *portmap* and *rpcbind* tools. Using RPC libraries in C/C++ or Java.
   - CORBA: server application, client application, interface definition in the IDL language. The use of CORBA

libraries in C/C++ or Java.

- RMI: server application, client application, definition of RMI services, registration of services in the RMI registry using the *rmiregistry* tool. The use of Java RMI libraries.
- JMS: client application of type of producer and consumer , JMS interface definition, 2 communication models between clients: point-to-point and publish-and-subscribe. The use of Java JMS libraries.
- WebServices: server application, client application, service definition using WSDL language, data exchange in XML format, use of SOAP protocol, registering a list of available services in the UDDI directory. Using libraries in C/C++, C# or Java.

4. Testing the call of remote services / procedures and the communication layer of the system

   Application testing in client-server and client-client architecture, depending on the chosen technology. Transmission of information between remote machines in point-to-point, point-to-multipoint and multipoint-to-multipoint systems.

5. Design and implementation of a graphical user interface for a remote service client.

   The form design and location of controls in the window (buttons, horizontal menu, edit windows, charts, etc.). Handling of dialogs for loading and saving data. Linking controls in a window with functions that call remote services/procedures.

6. Preparation of project documentation.

   The document confirming the implementation of the project includes: analysis of the problem, justification of choosing a particular technology in relation to the problem being solved, description of the most important parts of the client and server application and interface definition. Graphic representation of an example of calling remote procedures/services along with the flow of data between the points it concerns.

Recommended literature and teaching resources

1. F. Bolton, Pure CORBA, Pearson Education, 2001
2. R. Daigneau, Service Design Patterns - Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services, Pearson Education, 2011
3. W. Grosso, Java RMI, O'Reilly Media, 2001
4. M. Richards, R. Monson-Haefel, D. A. Chappell, Java Message Service. Creating Distributed Enterprise Applications, O'Reilly Media, 2009
5. J.R. Corbin, The Art of Distributed Applications, Springer-Verlag, 1991

Connection with area of study	engineering sciences

Student workload (ECTS credits balance)	
Student workload form	Student workload [hours]
Participation in project classes	30
Completion of a project	35
Individual consultations and final project presentation	10
Summary student workload	75
Module ECTS credits	
Workload of the direct assistance of the academic teacher	**1.6**
Workload of the practical classes	**3**

Annotation:
1 hour = 45 minutes