POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

COURSE DESCRIPTION CARD - SYLLABUS

Course name
Service-Oriented Architectures [S2Inf1-TPD>AZU]

Course

Field of study	Year/Semester
Computing	1/2

Area of study (specialization)	Profile of study
Data Processing Technologies	general academic

Level of study	Course offered in
second-cycle	Polish

Form of study	Requirements
full-time	compulsory

Number of hours

Lecture	Laboratory classes	Other
30	15	0

Tutorials	Projects/seminars	
0	15	

Number of credit points

5,00

Coordinators

dr inż. Juliusz Jezierski
juliusz.jezierski@put.poznan.pl

dr hab. inż. Maciej Zakrzewicz prof. PP
maciej.zakrzewicz@put.poznan.pl

Lecturers

Prerequisites

-

Course objective

1. Learn the fundamentals of designing enterprise application according to the Service Oriented Architecture. 2. Understand how to solve performance, availability and security problems of enterprise applications. 3. Learn teamwork skills.

Course-related learning outcomes

Knowledge:
has structured and theoretically well-developed general knowledge connected with the key issues of building and implementing applications of service-oriented architectures (k2st_w2)
has a theoretically founded detailed knowledge connected with selected issues in the field of computer science, such as: web services, business process automation, service buses (k2st_w3)

has advanced and detailed knowledge about the processes occurring in the service-oriented systems lifecycle (k2st_w5)
has knowledge about development trends and most significant new achievements in the area of technologies for implementation and deployment of web services (k2st_w4)

Skills:
is able to assess the usefulness and usability of web services (k2st_u6)
is able to - while formulating and solving tasks concerning development of applications of service-oriented architecture - integrate knowledge from different fields of computer science and apply a system approach, taking into account also non-technical aspects (k2st_u5)
is able to acquire information from literature, databases and other sources (in polish and english), integrate them, interpret and critically evaluate, draw conclusions and formulate and fully justify opinions (k2st_u1)
is able to assess the usefulness of methods and tools for solving an engineering task consisting in creation of a service-oriented application (k2st_u9)
is able - in accordance with a given specification, taking into account non-technical aspects - to design a complex device, information system or process, and to implement the project - at least in part - using adequate methods, techniques and tools, including adapting existing ones or developing new ones (k2st_u11)

Social competences:
understands that in computer science knowledge and skills become outdated very quickly, (k2st_k1)
understands the importance of using the latest knowledge of computer science in solving research and practical problems (k2st_k2)

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomespresented above are verified as follows:
- reports and assignments; includes teamwork
- final project presentation
- final test of choice - 30 single-choice questions, 15 correct answers required to pass

Programme content

Lectures:
Introduction to SOA: motivation, basic concepts, use cases. Service classification: business services, proxy services, process services, orchestration services, helper services. Overview of development technologies for SOAP Web Services. Review of the basic XML technologies: XML Schema, XPath, XSLT. BAsic standards and protocols: SOAP, WSDL, UDDI. Security standards: WS-Security, WS-SecurityPolicy, SAML. Java Enterprise Edition application server security features. Binary data transmission to/from Web Services: SOAP with Attachments, MTOM/XOP. Web Service Reliable Messaging (WSRM). Conversational Web Services. Development approaches for SOAP Web Services: Top-down, Bottom-up. Microservice Architecture concepts. RESTful Web Service concepts and implementation. Introduction to process and orchestration services. Business process modeling using BPEL. Implementing process services: business service invocation, security, transactions, adapters, business rules. Introduction to proxy services: ESB – Enterprise Service Bus, interaction models, security, SLA rules, monitoring. SOA deployment strategies, design patterns. Service-oriented analysis. SOA Governance.
Labs / project:
SOAP/RESTful Web Service modeling, design and development using Java Spring Boot, following the Top-down and Bottom-up approaches. Developing SOAP/RESTful Web Service clients. Using Apache Camel to orchestrate SOAP/RESTful Web Service invocations. Developing GUI for process services. Using message brokers for the Microservice Architecture. Deployment on containers. Database integration. Advanced final software projects.

Course topics

Teaching methods

Lecture: multimedia presentation, illustrated with examples given on the board.
Laboratory exercises: a multimedia presentation, a presentation illustrated with examples given on the board and tasks given by the lecturer, practical exercises.

Bibliography

Basic
1. Enterprise SOA: Service-Oriented Architecture Best Practices, Dirk Krafzig, Karl Banke, Dirk Slama, Prentice Hall PTR, 2004
2. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Thomas Erl, Prentice Hall PTR, 2004
3. BPEL Cookbook: Best Practices for SOA-based Integration and Applications Development, editors: Harish Gaur, Markus Zirn, PACKT Publishing, 2006
4. Service-Oriented Architectures: Concepts, Technology, and Design, Thomas Erl, Prentice Hall PTR, 2005
Additional

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	65	2,50