Course title: ADVANCED WEB TECHNOLOGIES AND SERVICES

Lecturers	Full Prof. Dragutin Kermek, Ph.D., Matija Novak, M.Inf.
Language of instruction:	Croatian and English
Schedule:	90 teaching hours - 15 hours per week (5 hours lectures + 10 hours laboratory exercises)
Study level	Master
Study programme	Information and Software Engineering
Semester	Summer
ECTS	7
Goal	Goal of the course is to introduce the students to advanced technologies which can help realization of complex Web applications. Upon completing this course the students will be able to individually develop advanced Web projects. In addition, they will be able to evaluate, master, install and use advantages of these packages. The course is project-oriented, in order to teach students how to plan, conduct and finish relatively big Java programming projects. The focus is on advanced Java programming applications where techniques (Applet, JSP, Servlet, XML, and Web services) can be used in combination with basic Java classes. The core of the course is program framework Java 2 Enterprise Edition (J2EE), used in construction of multi-layer, Web directed and complex applications. The course provides introduction to basic architecture underpinning J2EE, and deals with all the components of Java technologies contained in J2EE. In addition to understanding the purpose and importance of J2EE, students are given insight into focus points of their development plans. They also learn how to decide which technical skills the members of development team should have.
Content	1. **Introduction into Java programming language (6 hours)** History of Java programming language. Characteristics of Java programming language. Similarities with other programming languages. Program development. Platform independence. Version and edition types. Environment of Java programming language. Built-in data types. Operators. Instructions. Class definition. Method overlaps. Inheriting – specialization and generalization. Surpassing methods. Area of class and method application. Access to classes, methods and attributes. Interfaces and their role in

realization of abstract architectures. Grouping classes into packages. Creating class library. Built-in classes and interfaces. Java applet. JSDK examples.

2. **Advanced elements of Java programming language (4 hours)**

Covering mistakes and exceptions. Threads and multithreading Synchronization of threads. Outdated methods. Intern classes. Documenting programs. Models of managing graphic interface occurrences. U/I support. Access on byte and sign level. Using streams. Serializing data.

3. **Programming network resources (2 hours)**

Basic concepts of computer network. Internet standards. Determining computer names. Establishing connection between computers using sockets. Role of ports. Searching server's activities. Client testing. Realization of mini httpd server.

4. **Security (2 hours)**

Anatomy of Java program. Security models. Defining security policy. Defining access to local resources. Defining access to network resources. Defining access to outside clients. Transforming html data bank into jdk 2 format.

5. **Digital signature and certificate (2 hours)**

Bases of digital signature and certificate. Tools for document signing and creation of certificate. Steps in signing, taking over, sending and document receiving. Creating and installing signed applet. Creating and taking over the certificate.

6. **Program components (2 hours)**

Basic principles of component approach in program support development. Known architecture components.

JavaBeans component. Characteristics JavaBeans architectures: design patterns. Reflection and seriality of objects. Tools for creating of JavaBeans classes. Integration of new JavaBeans components into existing systems and tools. JAF – Java Activation Framework.

7. **J2EE elements for data access (4 hours)**

JDBC – Java DataBase Connectivity model. Types of JDBC managing programs. Realizations of connections to database. Execution of orders. Processing accepted data. Working with metadata. Group approach to database. XML. Defining structure: DTD and XML Scheme. Defining transformation of XML documents in other type of document using XSL. Processing XML documents using JAXP (Java API for XML Processing). Methods of processing XML documents: Simple API for XML - SAX and Document Object Model - DOM. Transformation architecture XSLT.

8. J2EE elements for servlet's side (6 hours)

Servlet's architecture. Servlet's life cycle. Classes and interfaces important for work with servlets. Initialization of parameter application and servlets. Tracking user's work using cookies and sessions. Communication between servlets and other resources. Supervision of user's work on level of application, session and some of their attributes. JavaServer Pages (JSP) architecture. Relation between JSP and servlets. Basic elements of JSP script language. Built-in objects. Expanding basic elements of JSP script language by applying libraries of JSP tags. Realization of user's interface using JavaFaces. Internationalization and localization of Web applications. Configuring delivery of Web application.

9. J2EE basic services (4 hours)

Bases of naming services. Naming services by using Java Naming and Directory Interface - JNDI. Bases of directory services. Directory services by using JNDI (Lightweight Directory Access Protocol - LDAP, Directory Naming Service - DNS, Network Information System - NIS, Novell Directory Service - NDS). Transactional service (Java Transactional API - JTA and Java Transactional Service - JTS). Messaging service (Java Messaging Service - JMS and Java API for XML Messaging - JAXM). Sending and receiving e-mail messages (JavaMail).

10. Distributed processing (4 hours)

Distributed systems. Evolution of distributed systems and architecture. CORBA architecture. Object Request Broker – ORB. Communication protocol for connecting CORBA applications through Internet (Internet-Inter-Orb Protocol – IIOP). Interface Definition Language – IDL. Development phases of CORBA program and needed tools. Remote Method Invocation – RMI. RMI architecture. RMI and IIOP. Copying Java into IDL. Development phases of RMI program and needed tools.

11. Web services (4 hours)

Architecture of Web service. Supporting standards for Web services. Simple Object Access Protocol – SOAP. Web Service Description Language – WSDL. Java API for XML based Remote Procedure Call - JAX-RPC. Universal

	Description, Discovery and Integration - UDDI. Comparing Web service to other architectures for remote processing. **12. J2EE complex component model (5 hours)** Technology of complex component model (Enterprise JavaBeans – EJB) and its benefits. When the application of EJB is recommended. Component for work tracking (Session Bean). Component for saving and data access (Entity Bean). Component for communicating (Message-Driven Bean). Realization of transactions. Access to resources. Security. Preparation for Web application delivery.
Exercises	In the course of the exercises the students use special program tools which are used for development of complex Web applications and Web services. They learn basic and advanced characteristics of Java programming language, characteristics of several component models, several ways in which Web applications can be realized. They also learn how to connect applications and distribute processing. To be able to realize the project the students need to learn to install and configure web servlet, application servlet and database servlet. Students are assigned project tasks to be completed and presented in set time.
Preconditions	-
Realization and examination	Classes: lectures, seminars and exercises Examination: homework assignments, project task and presentation
Related courses	1. Northern Michigan University, Advanced Web Programming, ttp://cs.nmu.edu/courses/cs460.html 2. Dalhousie University, Advanced Web Programming, http://www.cs.dal.ca/~jamie/course/CS/4173/ 3. Linköping University, Advanced Web Programming, http://www.ida.liu.se/education/ugrad/courses/tf/TDDB69/ 4. California State University, Web design and management, http://mieszko.csudh.edu/csc455su03.html
Literature	**Basic:** 1. Class materials available on closed system for e-education http://drava.foi.hr/fdl 2. Pat Niemeyer & Jonathan Knudsen: Learning Java, O'Reilly & Associates, Inc., 2000.

	3. Paul J. Perrone,Venkata S.R.R. Chaganti, Tom Schwenk, J2EE Developer's Handbook, Sams Publishing, 2003. 4. Scott Oaks, Java Security, O'Reilly, 2001. Additional: 1. Elliotte Rusty Harold, Java Network Programming, 2nd Edition O'Reilly & Associates, Inc, 2000 2. Jim Farley, Java Distributed Computing, O'Reilly, 1998. 3. Mark Wutka, Special Edition Using Java 2 Enterprise Edition, Que, 2001 4. Lajos Moczar & Jeremy Aston, Cocoon Developer's handbook, 2003.