

Study Program: INFORMATION TECHNOLOGY			
Level of Study: Undergraduate Vocational Studies			
Course Title: INTERNET PROGRAMMING			
Name of Lecturer: Mladen Janjić			
Course Language: English			
Semester: Winter			
ECTS: 6			
Prerequisites: Programming Languages, Object-Oriented Programming			
Course Objectives			
Proficiency with server-side (<i>backend</i>) programming, familiarity with methods and tools for development of dynamic web applications and services with application of software frameworks and contemporary technologies and tools. Understanding of multi-tier architecture, Model-View-Controller and other architectural models, developing Web API services and microservices with popular programming languages (<i>Python, JavaScript, C#, PHP, Java</i>).			
Learning Outcomes			
Students develop knowledge to create dynamic web applications, exchange data in JSON format between server and client using REST architecture, as well as Web API services and microservices. Through teamwork and application of code version management tools students acquire the knowledge necessary for effective work on projects in IT companies.			
Topics Summary			
<p><i>Lectures:</i> Server-side programming and currently popular programming languages for <i>backend</i> (server-side) applications, like <i>Python</i> (with <i>Flask</i>), <i>JavaScript</i> (with <i>Express.js</i>), <i>C#</i> (with <i>.NET</i>), <i>PHP</i> (with <i>Laravel</i>), or <i>Java</i> (with <i>Spring</i> and <i>Spring Boot</i>). Tools and standards for development of Web API server apps and exchanging data with client-side, handling GET, POST, PUT and DELETE requests, communicating with the database. Trends in dynamic web application development, REST architecture and RESTful application development, Web API services and microservices, 3-tier architecture. Using inversion of control, primarily dependency injection, Unit tests and mocking dependencies. Deployment of Web apps using automatization and CI/CD.</p> <p><i>Practical Classes:</i> Showcase of examples with web applications using a programming language of choice (e.g., <i>Python</i> with <i>Flask</i>). Using JSON format to exchange data with client-side apps. Developing web applications using a popular language framework and tools for installing libraries, testing and deployment. Mini-project – developing a web application in teams using the Github Classroom service and the Git tool for managing versions of program code, testing and deployment to Azure web service.</p>			
Readings			
<ol style="list-style-type: none"> 1. Miguel Grinberg, The New and Improved Flask Mega-Tutorial, Independently published, 2017, ISBN: 978-1977051875. (free online 2024 version with video: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world) 2. Al Sweigart, The Recursive Book of Recursion: Ace the Coding Interview with Python and JavaScript, No Starch Press, 2022, ISBN: 978-1718502024. (free online version: https://inventwithpython.com/recursion/) 3. Mark J. Price, C# 10 and .NET 6 – Modern Cross-Platform Development, Packt Publishing, 2021, ISBN: 978-1801077361. 4. Al Sweigart, Beyond the Basic Stuff with Python: Best Practices for Writing Clean Code, No Starch Press, 2020, ISBN: 978-1593279660. (free online version: https://inventwithpython.com/beyond/) 5. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd edition, O'Reilly Media, 2016, ISBN: 978-1491939369. (free online version: https://greenteapress.com/wp/think-python-2e/) 6. Allen B. Downey, Think Stats: Exploratory Data Analysis, 2nd edition, O'Reilly Media, 2014, ISBN: 978-1491907337. (free online version: https://greenteapress.com/wp/think-stats-2e/) 			
Active teaching hours: 4		Lectures: 2	
		Practical classes: 2	
Methods of instruction			
<p><i>Lectures:</i> Classical methods of teaching in combination with presentations and active interaction with students.</p> <p><i>Practical classes:</i> Examples that show step-by-step development of a web application with knowledge presented in lectures.</p>			
Examination methods (maximum 100 points)			
Exam prerequisites	50	Final exam	50
Activity during lectures	10		30
Project plan	40	Final exam	20

ECTS Grading system			
Local Grade	Description	Points	ECTS grade
10	Excellent	91 – 100	A
9	Exceptionally good	81 – 90	B
8	Very good	71 – 80	C
7	Good	61 – 70	D
6	Passing	51 – 60	E
5	Failing	0 – 50	F