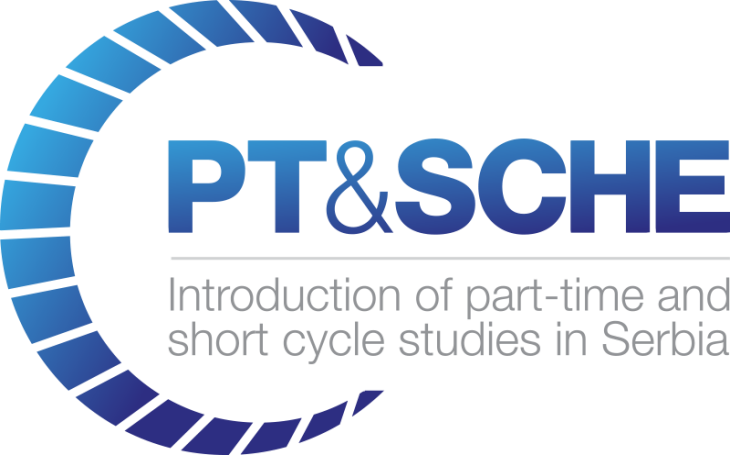
****

**D5.1.3**

**SHORT CYCLE PROGRAMME PROGRAMMING  
Modul: Web Application Programmer**

**Modul: Computer Programmer/Analyst**

|  |  |
| --- | --- |
| Project Acronym: | PT&SCHE |
| Project full title: | The Introduction of part‐time and short cycle studies in Serbia |
| Project No: | 561868-EPP-1-2015-1-EE-EPPKA2-CBHE-SP |
| Funding Scheme: | ERASMUS+ |
| Coordinator: | Tallinn University, TLU |
| Project start date: | October 15, 2015 |
| Project duration: | 36 months |

|  |  |
| --- | --- |
| Abstract | Computer programming is one of the most wanted and best-paying jobs today. The main objective of the short program Programming is to train the students for jobs: Web Application Programmer and Computer Programmer/Analyst. Both listed modules are the most popular in the IT market according to data from the appropriate sites at this moment. |

*"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsi­ble for any use which may be made of the information contained therein."*

DOCUMENT CONTROL SHEET

|  |  |
| --- | --- |
| Title of Document: | D5.1 Curriculum development of the short cycle program Programming |
| Work Package: | WP5. Pilot implementation of online PT & SCHE programs |
| Last version date: | 13/02/2018 |
| Status : | Final |
| Document Version: | 1.0 |
| File Name | Development of online SCHE Web Application Programmer |
| Number of Pages | 36 |
| Dissemination Level | Institutional |

VERSIONING AND CONTRIBUTION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Revision Description | Responsible Partner |
| 0.9 | 15.12.2017 | Concept development | Perica Štrbac, VIŠER |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[DOCUMENT CONTROL SHEET 2](#_Toc4495600)

[VERSIONING AND CONTRIBUTION HISTORY 2](#_Toc4495601)

[1 Introduction 5](#_Toc4495602)

[2 Program structure 6](#_Toc4495603)

[3 Subjects Syllabuses 8](#_Toc4495604)

[1.1. Computer Programmer/Analyst subjects 9](#_Toc4495605)

[Algorithms and data structures 9](#_Toc4495606)

[Relational database 10](#_Toc4495607)

[Soft computing 11](#_Toc4495608)

[Introduction to programming 12](#_Toc4495609)

[Object-oriented design 13](#_Toc4495610)

[Programming languages 14](#_Toc4495611)

[Object-oriented programming 2 15](#_Toc4495612)

[Software engineering 16](#_Toc4495613)

[Software testing 17](#_Toc4495614)

[Microprocessors software 18](#_Toc4495615)

[Introduction cloud computing 19](#_Toc4495616)

[Software technologies integration 20](#_Toc4495617)

[1.2. Web Application Programmer subjects 21](#_Toc4495618)

[Applicative software 21](#_Toc4495619)

[Database programming 22](#_Toc4495620)

[Web design 23](#_Toc4495621)

[Internet programming 24](#_Toc4495622)

[Internet protocols and technologies 25](#_Toc4495623)

[Internet services 26](#_Toc4495624)

[Web application programming 27](#_Toc4495625)

[Mobile devices programming 28](#_Toc4495626)

[Object-oriented programming 1 29](#_Toc4495627)

[Computer network security 30](#_Toc4495628)

[Standard user interfaces 31](#_Toc4495629)

[Visual programming techniques 32](#_Toc4495630)

[4 Comments on policy proposal of introduction of short cycle programs in Republic of Serbia 33](#_Toc4495631)

[1.3. SHORT CYCLE STUDY 33](#_Toc4495632)

[1 Duration od study 33](#_Toc4495633)

[2 Number of ECTS 33](#_Toc4495634)

[3 Recognition of ECTS gained on short cycle program for purposes of enrollment other study programs 33](#_Toc4495635)

[4 Quality 34](#_Toc4495636)

[5 Organizer and executor of short cycle program 34](#_Toc4495637)

[**6** Accreditation 34](#_Toc4495638)

[**7** National register of short cycle programs 34](#_Toc4495639)

[**8** Organization of teaching process 34](#_Toc4495640)

[9 Who can be a student of short cycle program? 35](#_Toc4495641)

[10 Teaching staff 35](#_Toc4495642)

[11 Preparation of curriculum of short cycle program and the supporting documents 35](#_Toc4495643)

[12 Approval of short cycle program 36](#_Toc4495644)

[13 Internship 36](#_Toc4495645)

[14 National certificate 37](#_Toc4495646)

[15 Partial realization of the program 37](#_Toc4495647)

[16 Financing of short cycle programs 37](#_Toc4495648)

# Introduction

Computer programming is one of the most wanted and best-paying jobs today. The main objective of the short program Programming is to train the students for jobs: WEB Application Programer and Computer Programmer/Analyst. Both listed modules are the most popular in the IT market according to data from the appropriate sites at this moment.

The program Web Application Programmer offers the following outcomes:

* Use fundamental skills to maintain web server services required to host a website.
* Select and apply markup languages for processing, identifying, and presenting of information in web pages.
* Use scripting languages and web services to transfer data and add interactive components to web pages.
* Create and manipulate web media objects using editing software.
* The program Web Application Programmer offers the following outcomes:
* Use fundamental skills to maintain web server services required to host a website.
* Select and apply markup languages for processing, identifying, and presenting of information in web pages.
* Use scripting languages and web services to transfer data and add interactive components to web pages.
* Create and manipulate web media objects using editing software.

The program Computer Programmer/Analyst offers the following outcomes:

* Apply programming processes within the programming language environment.
* Apply the basic concepts of object oriented programming, modularity, and structured code.
* Verify that the results obtained satisfy the original requirements.
* An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
* The program Web Application Programmer offers the following outcomes:
* Use fundamental skills to maintain web server services required to host a website.
* Select and apply markup languages for processing, identifying, and presenting of information in web pages.
* Use scripting languages and web services to transfer data and add interactive components to web pages.
* Create and manipulate web media objects using editing software.

# Program structure

Short cycle refers to as Programming includes two programs: Web Application Programmer and Computer Programmer/Analyst. Each program includes 12 elective subjects. Student chooses 5 subjects of 12 offered subjects. There is no one shared subject between these two programs.

Each subject amounts 6 ECTS and candidate should pass 5 subjects of 12 elective subjects per program. The volume of short cycle program refer to as Programming is 30 ECTS per program.

The curriculum of the short cycle program Programming is listed in the Table below. The table provides:

* subject’s name
* subject's position in the curriculum of the corresponding accredited program of basic studies in VISER;
* number of classes per week (classes of lectures+laboratory or auditory classes+additional forms of teaching);
* number of ECTS per subject based on corresponding accredited program of basic studies in VISER;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Subject Name** | **Sem.** | **Number of classes** | **ESPB** |
| **Computer Programmer/Analyst subjects** | | | | |
| **1.** | Algorithms and data structures | 2 | 2+3+0 | 6 |
| **2.** | Relational database | 6 | 2+3+0 | 6 |
| **3.** | Soft computing | 4 | 2+3+0 | 6 |
| **4.** | Introduction to programming | 2 | 2+3+0 | 6 |
| **5.** | Object-oriented design | 4 | 2+3+0 | 6 |
| **6.** | Programming languages | 3 | 2+3+0 | 6 |
| **7.** | Object-oriented programming 2 | 5 | 2+3+0 | 6 |
| **8** | Software engineering | 5 | 2+3+0 | 6 |
| **9.** | Software testing | 5 | 2+3+0 | 6 |
| **10.** | Microprocessors software | 6 | 2+3+0 | 6 |
| **11.** | Introduction cloud computing | 1 | 2+3+0 | 6 |
| **12.** | Software technologies integration | 6 | 2+3+0 | 6 |
| **Web Application Programmer subjects** | | | | |
| **1.** | Applicative software | 1 | 2+3+0 | 6 |
| **2.** | Database programming | 3 | 2+3+0 | 6 |
| **3.** | Web design | 3 | 2+3+0 | 6 |
| **4.** | Internet programming | 5 | 2+3+0 | 6 |
| **5.** | Internet protocols and technologies | 5 | 2+3+0 | 6 |
| **6.** | Internet services | 2 | 2+3+0 | 6 |
| **7.** | Web application programming | 5 | 2+3+0 | 6 |
| **8** | Mobile devices programming | 6 | 2+3+0 | 6 |
| **9.** | Object-oriented programming 1 | 4 | 2+3+0 | 6 |
| **10.** | Computer network security | 6 | 2+3+0 | 6 |
| **11.** | Standard user interfaces | 3 | 2+3+0 | 6 |
| **12.** | Visual programming techniques | 4 | 2+3+0 | 6 |

# 

# Subjects Syllabuses

Accreditation documentation of programs of basic studies in VISER (2017) contains the subjects syllabuses. Syllabuses of subjects that are planed for short cycle program Programming are listed below.

## Computer Programmer/Analyst subjects

#### Algorithms and data structures

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Algorithms and data structures | | | | | |
| **Lecturer:** Professor Svetlana Štrbac-Savić**,** PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computer literacy. | | | | | |
| **Aims:**  This module is designed to provide understandingof fundamental data structures and algorithms used in software development. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students gain fundamental problem solving skills and should be able to apply knowledge of algorithms and data structures in software development. | | | | | |
| **Module content:**  Lectures:   1. Basic data types. Simple and complex structures. 2. Static and dynamic structures. 3. Sequence-types and operations. Presentation of memory sequence. Optimization in sequence storing. 4. Lists, simple, double and circular lists linking - definition and operations. 5. Queues - definition, implementation and basic operations. 6. Stacks - definition, implementation and basic operations. 7. Trees - definition, representation and tree types. 8. Binary trees - definition, memory representation, basic operations. 9. Definition of the algorithm. Algorithm presentation . 10. Algorithic complexity. 11. Sorting. Linear complexity sorting methods. 12. Searhc. Sequencial and binary search. 13. Graphs, definitions, representations. Graph tour. Nodes reachability determination. 14. Compression. 15. Basic cryptographic methods.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. M. Tomašević (2005). Strukture podataka., Akademska misao. 2. A.Savić, S. Štrbac-Savić (2016): Priručnik za laboratorijske vežbe iz predmeta diskretna matematika i algoritmi, VIŠER, Beograd. 3. S. Đenić, J. Mitić, S. Štrbac (2006): Programiranje na jeziku C i osnovi programiranja na jeziku C++, zbirka primera i zadataka za laboratorijske vežbe iz predmeta osnovi programiranja 2. VIŠER, Beograd, 2006. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 5 | Written exam | 35 |
| Tutorials | | | 10 | Viva |  |
| Colloquia | | | 30 |  |  |
| Seminar paper(s) | | | 20 |  |  |

#### Relational database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Relational databases | | | | | |
| **Lecturer:** Professor Slobodan Obradović, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Database fundamentals and SQL literacy. | | | | | |
| **Aims:**  This module is designed to provide understandingof relational databases and their application in the information systems. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualifiedto design and implement relational databases. | | | | | |
| **Module content:**  Lectures:   1. Data models 2. Relational databases. 3. Relational algebra. 4. Normal forms: 1NF, 2NF, 3NF and Boyce-Codd normal forms. 5. Entity integrity, referential integrity, integrity limits. 6. Constraints. 7. Conceptual model. 8. Logic model. 9. Physical model. 10. Database re-engineering. 11. Standard modelling (pattern and standard models). 12. CASE tools. 13. Меtа моdelling. 14. Transaction processing system. 15. Physical database design.   Tutorial:  Working with databases in RDBMS Oracle 10g. | | | | | |
| **Readings:**   1. S. Obradović, S. Ilić (2016): SQL - Strukturirani upitni jezik za upravljanje relacionim bazama podataka, priručnik. VIŠER, Beograd. 2. H. Garcia-Molina, J. Ullman, J. Widow (2008): Database systems: The Complete Book. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Written exam | 60 |
| Practical lab-classes | | | 30 | Viva |  |
| Colloquia | | |  |  |  |
| Seminar paper(s) | | |  |  |  |

#### Soft computing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Soft computing | | | | | |
| **Lecturer:** Professor Nemanja Maček, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Math logic and basic algorithmic literacy. | | | | | |
| **Aims:**  This module is designed to provide understandingof soft computing and, in parts machine learning core concepts. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualifiedto apply core concepts of soft computing and machine learning. | | | | | |
| **Module content:**  Lectures:   1. Fundamentals of logic, artificial intelligence and soft computing. 2. Fuzzy logic. 3. Artificial neural networks. 4. Training, cross-validation and overfitting. 5. Deap learning. 6. Support Vector Machines. 7. Anomalies. 8. Evolutionary computing. 9. Genetic algorithms. 10. Artificial Immune Systems. 11. Game theory. 12. Chaos theory. 13. Rough sets. 14. Application of soft computing. 15. Application of machine learning.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. M. Milosavljević (2015): Veštačka inteligencija, Univerzitet Singidunum. 2. V. Miškovic (2013): Sistemi za podršku odlučivanju, Univerzitet Singidunum. 3. E. Volna (2013): Introduction to Soft Computing. Bookbon & Eva Volna 4. D. K. Pratihar (2013): Soft computing: fundamentals and applications. Alpha Science International, Ltd. 5. G. Tettamanzi, M. Tomassini (2013): Soft computing: integrating evolutionary, neural, and fuzzy systems. Springer Science & Business Media. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, term projects and viva. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Written exam | 30 |
| Practical lab-classes | | | 30 | Viva | 30 |
| Colloquia | | |  |  |  |
| Term project | | | 30 |  |  |

#### Introduction to programming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Introduction to programming | | | | | |
| **Lecturer:** Professor Slobodanka Đenić**,** PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computer literacy. | | | | | |
| **Aims:**  This module is designed to provide understandingof fundaments of structural programming and C programming language. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to design, develop and test programs in C programming language. | | | | | |
| **Module content:**  Lectures:   1. Introductory lecture. Basic concepts. 2. Development stages and program design methods. 3. Basic program structure. 4. Principal elements of C language. 5. Basic data types in C language. 6. Operators in C language programs. Functions from C library. 7. Order selection and loops in C language programs. 8. Order multiple selection and jumps in C language programs. 9. Numeric arrays in C langauge programs. 10. Character arrays in C language programs. 11. Sorting algorithms of arrays in C language programs. 12. Pointers and pointer application in C language programs. 13. Functions in C language programms. 14. Indicator application in functions in C language programms. 15. Overall discussion.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. L. Kraus (2014): Programski jezik C sa rešenim zadacima. Akademska misao, Beograd. 2. S. Đenić (2009): Osnovi programiranja 1. Elektronski udžbenik, VIŠER, Beograd. 3. A. Đenić, J. Mitić, S. Štrba (2009): Osnovi programiranja na jeziku C, zbirka primera i zadataka. VIŠER, Beograd. 4. S. Đenić, J. Mitić, S. Štrbac (2007): Rešeni zadaci na programskim jezicima C i C++, zbirka zadataka, VIŠER, Beograd. 5. S. Obradović (2004): Veština dobrog programiranja. VETŠ Beograd. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Practical exam | 30 |
| Tutorials | | | 10 | Viva |  |
| Colloquia | | | 30 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Object-oriented design

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Object-oriented design | | | | | |
| **Lecturer:** Professor Jelena Mitić**,** MSc | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Knowledge of basic object-oriented concepts. | | | | | |
| **Aims:**  This module is designed to provide understandingof object-oriented programming concepts and Java programming language. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop object-oriented programs in Java. | | | | | |
| **Module content:**  Lectures:   1. Data and expressions. 2. Classes and objects. 3. Encapsulation. 4. Conditions and loops. 5. Arrays. 6. Inheritance. 7. Polymorphism. 8. Working with databases. 9. Exceptions. 10. Recursion, 11. Collections. 12. UML. 13. Class diagrams. 14. Activity diagrams. 15. Fundamentals of softvare designing.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. Laslo Kraus (2015): Programski jezik Java sa rešenim zadacima JSE8. Akademska misao. 2. Bruce Eckel (2007): Misliti na Javi. Mikro knjiga. 3. John Lewis, William Loftus (2014): Java Software Solutions with Access Code: Foundations of Program Design. Pearson Education. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Practical exam | 40 |
| Tutorials | | | 10 | Viva | 10 |
| Colloquia | | | 40 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Programming languages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Programming languages | | | | | |
| **Lecturer:** Professor Slobodanka Mitić**,** MSc | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computer literacy. | | | | | |
| **Aims:**  This module is designed to provide understandingof advanced programming concepts in C and C++ programming languages. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to design, develop and test programs in C and C++ programming languages. | | | | | |
| **Module content:**  Lectures:   1. Introduction and basic concepts. 2. Functions, sequences and pointers in C. 3. Dynamic memory allocation in C. 4. Data structures in C. 5. Communication with operating system in C. 6. Working with files in C. 7. Preprocessing directives and modular programs in C. 8. Stacks, queues and dynamically linked list in C. 9. C elements in C++. 10. C inherited elements with features in C++. 11. Standard classes and objects in C++. 12. Data input and output, operators, sequences of commands and programs in C++. 13. Dynamic memory allocation in C++. 14. Working with files and dynamically linked lists in C++. 15. Overall discussion, self-evaluation.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. L. Kraus (2014). Programski jezik C sa rešenim zadacima. Akademska misao, Beograd. 2. L. Kraus (2016). Programski jezik C++ sa rešenim zadacima. Akademska misao, Beograd 3. S. Đenić (2014): Programski jezici, elektronski udžbenik. VIŠER, Beograd. 4. S. Đenić, J. Mitić, S. Štrbac (2009): Programiranje na jeziku C i osnovi programiranja na jeziku C++, zbirka primera i zadataka. VIŠER, Beograd. 5. S. Đenić, J. Mitić, S. Štrbac (2007): Rešeni zadaci na programskim jezicima C i C++, zbirka zadataka. VIŠER, Beograd. 6. Broj časova aktivne nastave | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Practical exam | 30 |
| Tutorials | | | 10 | Viva |  |
| Colloquia | | | 30 |  |  |
| Seminar paper(s) | | | 20 |  |  |

#### Object-oriented programming 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Object-oriented programming 2 | | | | | |
| **Lecturer:** Professor Perica Štrbac, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Knowledge of basic object-oriented concepts. | | | | | |
| **Aims:**  This module is designed to provide in-depth understanding of object-oriented programming concepts and Java programming language. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop object-oriented programs in Java. | | | | | |
| **Module content:**  Lectures:   1. Classes. Inheritence, abstract classes, interfaces. 2. Polymorphism, exceptions.. 3. Class Object, interface, constructor, static initialization blocks. 4. Standard library. Input/oputput. Serialization. 5. Concurrent programming. Threads. 6. Sinhronitazation. 7. Thread groups. 8. GUI: АWТ 9. GUI: Swing 10. GUI: JavaFX. 11. Network programing, client-server architecture. 12. Using TCP/IP, URL and URLConnection in Java. 13. File transfer in client-server communication. Communication via datagrams. 14. Working with databases. 15. Multilayered system technologies.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. Yakov Fain (2015): Java 8 programiranje. Mikro knjiga. 2. Bruce Eckel (2014): Misliti na Javi, prevod 4. izdanja. Mikro knjiga. 3. Cay S. Horstmann and Gary Cornell (2013): Core Java, Advanced Features, Vol. 2. Prantice Hall. 4. Cay S. Horstmann and Gary Cornell (2011): JAVA 2 Advanced Features, Prantice Hall. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Written exam | 30 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | | 50 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Software engineering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Software engineering | | | | | |
| **Lecturer:** Professor Goran Šimić, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Knowledge of basic object-oriented concepts. | | | | | |
| **Aims:**  This module is designed to provide introduction to software engineering, theory and practical stages during software development process in all phases of its life-cycle. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop software systems beased on tools explained in the course of the module. | | | | | |
| **Module content:**  Lectures:   1. Software and software engineering. Basic concepts and fundamental paradigms. 2. The life cycle of software development and "prototyping". 3. Software project management. 4. Metrics for Evaluating Policy Commitment 5. Planning, documentation 6. UML as a visual modeling language. 7. CASE tools. 8. Software requirements and system analysis. 9. Software system architecture design 10. Software design and implementation 11. Software development process. 12. Software validation and verification. 13. Software testing techniques and strategies. 14. Software Maintenance. 15. Software quality assurance   Tutorial:  Tuition is in compliance to the curriculum. Demonstration of practical software system development by techniques presented during lactures, using selected CASE tools. Curricula in compliance to te recommendations of IEEE/ACM ComputingCurriculum: CE2004 Computer Engineering Body of Knowledge: CE-SWE 0-9. | | | | | |
| **Readings:**   1. I. Sommerville (2000): Software Engineering, 6th ed. Addison-Wesley, Reading, MA. 2. R.S. Pressman (2001): Software Engineering: A Practitioner's Approach. McGraw Hill, NY, 5th ed.. 3. M. Fowler, K. Scott (1999): UML Distilled: A Brief Guide to the Standard Object Modeling Language, 2nd ed. Addison-Wesley, Reading, MA. 4. G. Booch (1994): Object-Oriented Analysis and Design with Applications, 2nd ed. Addison-Wesley, Reading, MA. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Written exam | 30 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | | 50 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Software testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Software testing | | | | | |
| **Lecturer:** Professor Jelena Mitić, MSc | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic programming literacy. | | | | | |
| **Aims:**  This module is designed to provide introduction to software testing during software development life-cycle, including, but not limited to black box, white box and fuzzy testing approach as well as with some best practices. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to select and apply appropriate software testing technique. | | | | | |
| **Module content:**  Lectures:   1. Introduction to software testing. 2. Manual and automatic testing. 3. Black box approach. 4. White box approach. 5. Data flow approach. 6. Fuzzy software testing. 7. Integral and regressive testing. 8. Testing object-oriented software. 9. System testing. 10. Testin as a software development phase. 11. Criteria selection. 12. Theoretical and practical testing limitations. 13. Testing tools, part 1. 14. Testing tools, part 2. 15. Overall discussion, self-evaluation.   Tutorial:  Tuition is in compliance to the curriculum. Demonstration of practical software system development by techniques presented during lactures, using selected CASE tools.  Curricula in compliance to te recommendations of IEEE/ACM ComputingCurriculum: CE2004 Computer Engineering Body of Knowledge: CE-SWE 0-9. | | | | | |
| **Readings:**   1. Jovan Popović (2012): Testiranje softvera u praksi. Mikroknjiga. 2. Rex Black, Erik Van Veenendaal, Dorothy Graham (2012): Foundations of Software testing. Cengage Learning 3. A. Mathur (2014): Foundations of Software Testing. Addison-Wesley Professional. 4. Hariprashath P (2015): Software testing: 100+ Testing Approaches. Kindle eBook. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Written exam | 40 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | | 40 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Microprocessors software

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Microprocessor software | | | | | |
| **Lecturer:** Professor Milan Mijalković, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic programming literacy and PC architecture. | | | | | |
| **Aims:**  This module is designed to provide introduction to practical application of microcontroller programming in C (and partially, in assembler) as well as to develope basic practical programming skils and techniques in application of modern microcontroller peripherals. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to design simple programmes used in practical tasks in embedded computers, use program development software tools, and to use integrated development environments (IDE) for microcontrollers. | | | | | |
| **Module content:**  Lectures:   1. Microcontroller architecture and the role of individual components. Principles of assembler. 2. Microcontroller overview, modular programming, segmentation, linker. 3. Content and design concept, role and usage of certain development tools. 4. Creating of standard programming structures for specific microcontroller. Macros. 5. Interrupt mechanism. Real time operating systems, context switching. 6. General purpose IO (GPIO), AD and DA converters, timers and counters. 7. Specifics of the system with reduced power and increased reliability. 8. Examples of applicable programs. Look-up tables, mathematical functions. 9. Application in control. Filtering, real-time signal generation, PWM. 10. Signal measurement and digitizing, Delay impact. Influence of finite word length data. 11. Advanced techniques. Multiprocessor systems, networking, CAN, Zigbee. 12. Specificities of C- language for microcontrollers and differences from standard ANSI C. 13. Design methods of embedded microcomputers. Teamwork, documentation. 14. Tools for programming development. Simulators, monitor, emulators, logic analyzers. 15. Overall discussions. Further professional development.   Tutorial:  Tuition is in compliance to the curriculum. Students independently develop, translate and test accuracy of its program on real hardware in the lab.  Module in compliance to the recommendations of IEEE/ACM Computing Curriculum: CE2004 Computer Engineering Body of Knowledge: CE-ESY. | | | | | |
| **Readings:**   1. M. Mijalković, Ž. Popov (2003): Priručnik za laboratorijske vežbe iz mikroprocesorskog softvera. VETŠ, Beograd. 2. NXP, LPC2138 User's Manual, Nxp (2014). 3. IAR, IAR Embedded Workbech for ARM User’s Manual (2013). | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Written exam |  |
| Practical lab-classes | | | 50 | Viva | 40 |
| Colloquia | | |  |  |  |
| Seminar paper(s) | | |  |  |  |

#### Introduction cloud computing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Introduction to Cloud Coputing | | | | | |
| **Lecturer:** Professor Borislav Đorđević, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Operatings system and networking literacy. | | | | | |
| **Aims:**  This module is designed to provide introduction to fundamental cloud computing concepts. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to use modern technologies within the cloud computing domain. | | | | | |
| **Module content:**  Lectures:   1. Computer architecture fundamentals. 2. Operating system fundamenals. 3. Computer network and security fundamentals. 4. Introduction to virtualization. 5. Types of virtualization. 6. Server virtualization and desktop virtualization. 7. Introduction to Cloud Computing. 8. Cloud Computing models. 9. Software as a Service. 10. Google Apps, Microsoft Office 365. 11. Platform as a Service. 12. Google App Engine, Microsoft Azure 13. Infrastructure as a Service. 14. Amazon CloudFormation (EC2). 15. Overall discussion.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. A. Silberschatz, P. Galvin, G. Gagne (2009): Operating System Concepts, 8th edition. John Wiley&Sons, Inc. 2. B. Đorđević, D. Pleskonjić, N. Maček (2006): Operativni sistemi: teorija praksa i rešeni zadaci. Mikro Knjiga, Beograd. 3. Thomas Erl, Ricardo Puttini, Zaigham Mahmood (2014): Cloud Computing: Concepts, Technology & Architecture. Prentice Hall. 4. Matthew Portnoy (2012): Virtualization Essentials. Sybex. 5. Gustavo A. A. Santana (2014): Data Center Virtualization Fundamentals. Cisco System Inc. 6. Rogier Dittner, David Rule Jr. (2007): The Best Damn Server Virtualization Book Period. Elsevier Inc. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Written exam | 70 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | |  |  |  |
| Seminar paper(s) | | |  |  |  |

#### Software technologies integration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Computer programmer / analyst | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Software technologies integration | | | | | |
| **Lecturer:** Professor Zoran Ćirović, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Object-oriented programming and Internet literacy. | | | | | |
| **Aims:**  This module is designed to provide introduction to software technologies integration. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to design and develop software using diverse programming platforms and operating systems. | | | | | |
| **Module content:**  Lectures:   1. Introduction. 2. XML fundamentals. 3. XML data validation: DTD и XML scheme. 4. Parsing XML documents 5. XPATH and XSLT technologies and data set transformation. 6. XQuery, XPointer, XLink. 7. Scriptong languages.. 8. Application of scripting languages.. 9. Review. 10. XML services: endpoint, service contract type and connections. 11. XML services: practical implementation and disclosure. 12. Resource architecture and REST service design. 13. Service to system data protection. 14. Diagnostics. 15. Overall discussion.   Tutorial:  Tuition is in compliance to the curriculum.. | | | | | |
| **Readings:**   1. W.S.Means (2006): E.R.Harold, XML za programere. Mikro knjiga. 2. L. Lemay, R. Colburn, J. Kyrnin (2016): HTML5, CSS3 i JavaScript za razvoj web strana. Kompjuter biblioteka. 3. J. Lowy, M. Montgomery (2015): Programming WCF Services, 4th Edition. O'Reilly Media. 4. J. Webber, S. Parastatidis, I. Robinson (2010): REST in Practice. O'Reilly Media. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Written exam | 50 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | |  |  |  |
| Seminar paper(s) | | | 20 |  |  |

## Web Application Programmer subjects

#### Applicative software

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Application software | | | | | |
| **Lecturer:** Professor Jelena Mitić**,** MSc | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computers skills and literacy, knowledge of operating systems and file management. | | | | | |
| **Aims:**  This module is designed to provide necessary understandingof standard application software usage and introduction to digital society. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be ableto understand principles of application software and qualified to use word processors, spreadsheets, presentation design software as well as basic Internet services. | | | | | |
| **Module content:**  Lectures:   1. Introductory lecture (the organization and content of the module). Мicrosoft Office, Libre Office. 2. Basic word processing techniques. 3. Entering and editing text, formatting text, characters, paragraphs and pages. 4. Advanced word processing techniques. Embedding objects in text. 5. Tables. Equation editor. 6. Processing longer texts. Styles. 7. Spreadsheets. Basic concepts. 8. Edit cell content, editing a worksheet.. 9. Formatting spreadsheets. Examples. 10. Functions, basic application techniques. 11. Diagrams. Databases. Advanced techniques. 12. Presentations. Basic rules presentation creation and display techniques. Slide creation. 13. Presentations. Handling slide components. Installation of object. Animation. 14. Internet services. Web search engines, e-mail, discussion lists, publishing presentations on the Internet. 15. Combined use of different programs.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. R. Vukić, D. Perić, I. Vlajić-Naumovska (2011): Aplikativni softver, 3. izdanje. VIŠER, Beograd. 2. ECDL official literature: modules 3,4,6 and 7. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Written exam | 40 |
| Tutorials | | |  | Viva |  |
| Colloquia | | | 20 |  |  |
| Seminar paper(s) | | | 40 |  |  |

#### Database programming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Database programming | | | | | |
| **Lecturer:** Professor Slobodan Obradović, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computers skills and literacy, knowledge of operating systems and file management. | | | | | |
| **Aims:**  This module is designed to provide necessary understandingof databases and SQL queries, their application in the information systems, and the MS Access database management system. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualifiedto design relational databases, develop accompanying applications and use SQL queries over complex relational databases. | | | | | |
| **Module content:**  Lectures:   1. Data models. 2. Relational databases. 3. Primary and forein key, key constraints. 4. Database management systems. 5. Fundamentals of query langauges. 6. Relational algebra. 7. SQL and QBE query languages. 8. Aggregate functions. 9. Parametrized queries. 10. Permissions and roles. 11. Stored procedures and triggers. 12. Designing relational databases 13. 1NF, 2NF, 3NF and Boyce-Codd normal forms. 14. Transaction processing system. 15. Physical database design.   Tutorial:  Designing databases and accompanying applications with Microsoft Access. | | | | | |
| **Readings:**   1. P. Kaluđerčić, S. Obradović (2015): Projektovanje informacionih sistema i relacione baze podataka). VIŠER, Beograd, 2015. 2. S. Obradović, T. Pandurov, B. Bojičić (2015): SQL struktuirani upitni jezik. VIŠER, Beograd, 2015. 3. S. Obradović, B. Pavić, V. Petković, G. Dimić (2015): MS Access 2013 – projektovanje baza podataka i aplikacija. VIŠER, Beograd, 2015. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Written exam | 35 |
| Practical lab-classes | | | 20 | Viva | 35 |
| Colloquia | | |  |  |  |
| Seminar paper(s) | | |  |  |  |

#### Web design

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Web design | | | | | |
| **Lecturer:** Professor Branimir Trenkić, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computer skills and literacy. | | | | | |
| **Aims:**  This module is designed to provide necessary understandingof Web site design, development and publishing. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualifiedto design, develop, publish and maintain commercial Web sites. | | | | | |
| **Module content:**  Lectures:   1. Content, technology, visual elements and efficiency in Web design. 2. Elements of good design. Similarities and differences between Web design and periodical publication design and electronic media. Developing and publishing costs. 3. Website design process. Storyboard. 4. Development and check-up of a background Web server prototype. Publishing site. 5. Conventions on the Web. Static and dynamic Web locations. 6. Web site organization. Parent page, interactive pages, descriptive pages, press-optimized pages, TLB pages, orphan pages and pages with copyright statement. 7. Navigation system. Primary and secondary navigation system. 8. Bookmark of text pages. Links. Icons and mapped images. 9. Page label. Depth meter. Graphic landmarks. 10. SEO optimization. Local browser and site map. 11. Text as the basic expression tool. System fonts and Internet fonts. 12. Multimedia environment. Colours and their meanings. Bitmap and vector images supported on the Web. 13. Animation and sound. Web-supported formats. 14. CMS Web systems. Joomla. WordPress. 15. Domain selection. Hosting service. Delivery and maintenance of web location..   Tutorial:  Designing Web site with DreamWeaver, Adobe Photoshop and Corel Draw. | | | | | |
| **Readings:**   1. P. Staletić (2015): Priručnik iz Web dizajna. VIŠER, Beograd, 2015. 2. T. Powell (2002): Web dizajn, kompletan priručnik. Mikroknjiga, Beograd, 2002. 3. C. Eccher (2008): Professional Web Design, Third Edition, Cengage Learning, 2008. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, term projects and viva. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Written exam |  |
| Practical lab-classes | | | 20 | Viva | 30 |
| Colloquia | | |  |  |  |
| Term project | | | 40 |  |  |

#### Internet programming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Internet programming | | | | | |
| **Lecturer:** Professor Boško Nikolić, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Familiarity with basic Internet application programmimg technologies. | | | | | |
| **Aims:**  This module is designed to provide necessary understandingof modern Internet applications design and development using Java programming language. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to design and develop modern Internet applications using Jаva programming language. | | | | | |
| **Module content:**  Lectures:   1. Basic concept and terms. 2. Internet protocols. 3. Architecture of multi-layer client-server systems 4. Properties and structure of Internet applications. 5. Page description languages. XML. HTML. DOM. JavaScript. 6. Server-side scripting. 7. Web application development platform. 8. Java applications, applets. 9. Servlets. 10. Working with databases. JDBC. 11. STRUTS framework. 12. Web forms and user interraction. 13. JCF. 14. Web services. 15. Security issues: session authentication and authorization, encryption, public key infrastructure, HTTPS.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. B. Nikolić (2006): Programiranje Internet aplikacija. Fakultet za poslovnu informatiku, Beograd. 2. B. Nikolić (2008): Programiranje Internet aplikacija. Beograd. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, seminar papers, and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Written exam | 60 |
| Practical lab-classes | | | 10 | Viva |  |
| Colloquia | | |  |  |  |
| Seminar paper(s) | | | 30 |  |  |

#### Internet protocols and technologies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Internet protocols and technologies | | | | | |
| **Lecturer:** Professor Branimir Trenkić, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Computer networking skills and literacy. | | | | | |
| **Aims:**  This module is designed to provide necessary understandingof local computer networking technology, inter-connected computer networks, protocols and routing algorhytms. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to desing the network and fine-tune network device parameters (networked computers and interconnection devices, such as switches, bridges and routers). | | | | | |
| **Module content:**  Lectures:   1. Layered protocol architecture. Protocols, services, primitives, moulding. 2. Mutual comparison of OSI and TCP/IP stacks. 3. Link layer protocols: CSMA / CD, CSMA / CA, PPP. 4. IPv4 protocol. IP addresses, subnets, protocol formats and mechanism, network address translation. 5. IPv6 protocol. 6. Routing. RIP and OSFR protocols. 7. Transport layer. TCP protocol. 8. Congestion management algorithms. Tahoe, Reno. 9. Carnot's algorithm; congestion avoidance algorithm, ACK clock. 10. Simple network management control (SNMP). 11. File Transfer Protocol (FTP), HTTP protocol. 12. E-mail (SMTP protocol). Domain Name System (DNS). 13. Multimedia system. Standards: H323, SIP. 14. Broad area network architecture. Frame relay, ATM, Multiprotocol label switching (MPLS), private virtual network. 15. Wireless Communication Systems: IEEE802.11 (WiFi), IEEE802.15 (Bluetooth, ZigBee), IEEE802.16, (WiMax).   Tutorial:  Iinstallation and tunning of networking operating systems, establishing connection with network devices, router configuration and fine-tuning.  Module is in compliance to the recommendations of IEEE/ACM Computing Curriculum: CE2004 Computer Engineering Body of Knowledge: CE -NWK 0-5. CC2008 Computer Science Body of Knowledge: CC-NC 2, CC-NC 4, CC-NC 6, CC-NC 8. | | | | | |
| **Readings:**   1. V. Vasiljević (2013): Internet protokoli i i tehnologije. VIŠER, Beograd. 2. V. Vasiljević, P. Gavrilović, B. Krneta, M. Kamberović (2016): Internet protokoli i tehnologije – priručnik. VIŠER, Beograd. 3. W. Stallings (2014): Data and Computer Communications. Pearson Ed., Inc. Upper Saddle Riv., NJ, 2014. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Practical exam | 30 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia or quizzes | | | 20 |  |  |
| Seminar paper(s) | | | 20 |  |  |

#### Internet services

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Internet services | | | | | |
| **Lecturer:** Professor Branimir Trenkić, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computer skills and literacy. | | | | | |
| **Aims:**  This module is designed to provide necessary understandingof the most important Internet services. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to use the most important Internet services. | | | | | |
| **Module content:**  Lectures:   1. Internet fundamentals. 2. Connecting to the Internet. Dial-Up connection. DSL connection. 3. Broadband internet. Access via GPRS and G3 (UMTS) mobile phone networks. Wireless access. 4. URL. Domains. registration to international and local domains. 5. Internet syntax. 6. E-mail service. 7. Internet protection. 8. Web service. Browsing the Web. Advanced Google search. 9. On–line shopping, travel and economies on the Internet. 10. Multimedia on the Internet. Photos, music and sounds, videos, movies and TV. 11. Social networks. Facebook and Twitter. 12. Client relationship management via Facebook and Twitter. 13. Blog. Maintaining public relations via blogging. 14. Internet telephony (VoIP). Similarities and differences between landline and Internet telephony, 15. Equipment for internet telephony. Internet clients Skype, Viber, Webex.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. P. Staletić (2001): Internet servisi u poslovnoj komunikaciji. VETŠ, Beograd. 2. P. Staletić (2016): Priručnik iz Internet servisi u poslovnoj komunikaciji. VIŠER, Beograd. 3. M. Jakobsson (2012): The Death of The Internet. Wiley & Sons. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Writen exam |  |
| Practical lab-classes | | | 20 | Viva | 50 |
| Colloquia | | | 20 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Web application programming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Web application programming | | | | | |
| **Lecturer:** Professor Milan Gnjatović, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computer skills and literacy. | | | | | |
| **Aims:**  This module is designed to provide necessary understanding of the basics of web programming and qaulifiying students to design and develop Web-oriented PHP applications. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop Web-oriented applications using PHP. | | | | | |
| **Module content:**  Lectures:   1. Saving and loading data, the use of sequences. 2. Working with strings and regular expressions, writing functions. 3. Object-oriented PHP, errors and exceptions handling. 4. Desinging and developing database for the Web application. 5. Using MySQL. 6. Advanced MySQL. 7. User authentication and Web application protection. 8. Secure transactions. 9. File system and server operations using network functions and protocols. 10. Operations on dates, periods and images. 11. Session management in PHP and debugging. 12. Using PHP and MySQL for large projects. 13. Web services linking. 14. Ajax in web application design. 15. CMS basic concepts.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. M. Gnjatović (2017): Uvod u pronalaženje informacija na vebu. VIŠER, Beograd. 2. L. Weilling, L. Thomson (2009): PHP i MySQL – razvoj aplikacija za Web, prevod 4. izdanja. Mikro knjiga, Beograd. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Writen exam | 40 |
| Practical lab-classes | | | 10 | Viva | 10 |
| Colloquia | | | 40 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Mobile devices programming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Mobile devices programming | | | | | |
| **Lecturer:** Professor Nemanja Maček, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Knowledge ofJava and XML. | | | | | |
| **Aims:**  This module is designed to provide necessary understanding of the basics of mobile application development for Android devices. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop Android applications applications. | | | | | |
| **Module content:**  Lectures:   1. Introduction to Android operating system. 2. MIT App Inventor. 3. Fundamentals of Android application development: Java. 4. Fundamentals of Android application development: JSON and XML. 5. Activities, fragments and intents. 6. User interface. 7. Designing user interface with views.Дизајнирање кориснићког интерфејса применом погледа 8. Images and menus. 9. Data persistence. 10. Content providers. 11. Broadcast recievers. 12. Location based services. 13. Networking. 14. Android services development. 15. Publishing the application.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. Wei-Meng Lee (2013): Android 4 razvoj aplikacija, drugo izdanje. Kompjuter biblioteka, Beograd. 2. Paul Deitel, Harvey Deitel, Alexander Wald (2016): Android 6 for Programmers: An App-Driven Approach. Deitel, 2016. 3. Derek Walter, Mark Sherman (2014): Learning MIT App Inventor: A Hands-On Guide to Building Your Own Android Apps. Addison-Wesley. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Writen exam |  |
| Practical lab-classes | | |  | Viva | 10 |
| Colloquia | | | 40 |  |  |
| Seminar paper(s) | | | 40 |  |  |

#### Object-oriented programming 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Object-oriented programming 1 | | | | | |
| **Lecturer:** Professor Perica Štrbac, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Knowledge of C programming language. | | | | | |
| **Aims:**  This module is designed to provide necessary understanding of the basics of basics of the object-oriented programming and qualifiying the students to develop complex object-orinted programs in the C++ programming language. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop develop complex object-orinted programs in the C++ programming language. | | | | | |
| **Module content:**  Lectures:   1. Core concepts of object oriented programming. 2. Non-object oriented elements of C++. 3. Classes. Objects. Constructors and destructors. 4. Class members pointers. 5. Operator functions. 6. Inhertitence. 7. Derived classes. 8. Constructors and destructors of derived classes. 9. Virtual destructors. Virtual classes. 10. Generic functions and classes. 11. Exceptions. 12. The standard class library. 13. Threads. 14. Iput / output. 15. Lambda.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. L. Kraus (2011): Programski jezik C++ sa rešenim zadacima, osmo izdanje. Akademska misao, Beograd. 2. B. Stroustrup (1997): The C++ Programming Language. 3. B. Eckel (2000): Thinking in C++, Planet PDF. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Writen exam | 30 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | | 50 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Computer network security

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Computer network security | | | | | |
| **Lecturer:** Professor Nemanja Maček, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic knowledge of operating systems, computer networks and programming. | | | | | |
| **Aims:**  This module is designed to provide necessary understanding of the basics of computer systems and network security as well as to qualify students to identify threats and subsequentially employ security mechanisms and services. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to identify threats and vulnerabilities and employ and fine-tune various security mechanisms and services. | | | | | |
| **Module content:**  Lectures:   1. Core concepts of computer security 2. Security architecture and models. 3. Cryptography. 4. Cryptographic protocols. 5. Access control and firewalls. 6. Intrusion detection systems. 7. Malware. 8. E-commerce and Internet security. 9. Wireless and mobile networks security. 10. Operating systems security. 11. Database security. 12. Secure coding. 13. Computer network monitoring. 14. Disaster recovery and business continuity planning. 15. Ethical hacking and penetration testing   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. G. Grubor, M. Milosavljević (2010): Osnove zaštite informacija, Univerzitet Singidnum, Beograd. 2. A. Jevremović, M. Veinović, M. Šarac, G. Šimić (2014): Zaštita u računarskim mrežama. Univerzitet Singidunum, Beograd. 3. D. Pleskonjić, N. Maček, B. Đorđević, M. Carić (2007): Sigurnost računarskih sistema i mreža. Mikro knjiga, Beograd | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Writen exam | 30 |
| Practical lab-classes | | | 30 | Viva |  |
| Colloquia | | | 30 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Standard user interfaces

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Standard user interfaces | | | | | |
| **Lecturer:** Professor Perica Štrbac, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Basic computer skills and literacy. | | | | | |
| **Aims:**  This module is designed to provide necessary understanding of the basics of HTML5, CSS3 and JavaScript technologies as well as to qualify students to solve client side programming problems. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop interactive client-side programs using HTML5, CSS3 and JavaScript technologies. | | | | | |
| **Module content:**  Lectures:   1. HTML5. 2. CSS3. 3. JavaScript: variables, conditions, loops, funcions, objects, events, selectors, interraction with the documents. 4. Forms interface. 5. Audio and video interfaces. 6. Fullscreen and Strean interfaces. 7. Canvas and SVG interfaces. 8. PointerLock and DragandDrop interfaces. 9. WebStorage and Geolocation interfaces. 10. History and Offline interfaces. 11. PageVisibility and WebMessaging interfaces. 12. WebSocket and WebWorkers interfaces. 13. File interface. 14. Ajax level 2. 15. Angular 2.   Tutorial:  Tuition is in compliance to the curriculum. | | | | | |
| **Readings:**   1. Juan Diego Gauchat (2014): Integrisane tehnologije za izradu Web strana, drugo izdanje. Mikro knjiga, Beograd. 2. D, Flanagan (2008): JavaScript: sveobuhvatni vodič. Mikro knjiga, 2008. 3. Pablo Deeleman (2016): Naučite Angular 2. Kompjuter biblioteka. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | |  | Writen exam | 30 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | | 50 |  |  |
| Seminar paper(s) | | |  |  |  |

#### Visual programming techniques

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short cycle:** Web Application Programmer | | | | | |
| **Type and level of the module:** Undergraduate module | | | | | |
| **Module title:** Visual programming techniques | | | | | |
| **Lecturer:** Professor Zoran Ćirović, PhD | | | | | |
| **Module status:** Optional | | | | | |
| **ECTS credits:** 6 | | | | | |
| **Prerequisites:** Understanding the core concepts of object-oriented programming | | | | | |
| **Aims:**  This module is designed to provide necessary understanding of the concepts of visual programming as well as to qualify students to develop applications in C#. | | | | | |
| **Learning outcomes:**  On successful completion of this module, students should be qualified to develop applications using C#. | | | | | |
| **Module content:**  Lectures:   1. Applications performances under new operating systems. 2. Data sharing between program units and modules. 3. .NET platform. Fundamentals of C # programming language. 4. Classes and interfaces. 5. Designing the graphical user interface (GUI). 6. Visual object linking and embedding. 7. Programming language integration. 8. Resources and I/O devices management. 9. Multimedia content. 10. Windows, multiple document support, dynamic libraries. 11. Literature utilization / support guidelines. 12. Forms, printing, context-sensitive help. 13. XML support. 14. Fundamentals of application-related data sources. 15. Web services. Interface concept.   Tutorial:  Tuition is in compliance to the curriculum.  . | | | | | |
| **Readings:**   1. J. Allwork (2016): C# Programiranje za Windows i Android. InfoElektronika 2. M. Price (2016): C# 6 i .NET Core 1.0 moderno međuplatformsko programiranje, Kompjuter biblioteka. 3. J. Albahari, B. Albahari (2015): C# 5.0 za programere – sveobuhvatan referentni priručnik, Mikro knjiga, Beograd. | | | | | |
| **Teaching units: 75** | | | | | **Other classes** |
| Lectures: | Tutorials: | Other forms of teaching: | | Research study: |  |
| **2** | **3** |  | |  |
| **Teaching methods:**  Lectures, practical lab classes, consultation, colloquia, seminar papers and written exam. | | | | | |
| **Assessment methods (maximum number of points 100)** | | | | | |
| **Pre-exam obligations** | | | Points | **Final exam** | Points |
| Student’s engagement | | | 10 | Writen exam | 50 |
| Practical lab-classes | | | 20 | Viva |  |
| Colloquia | | |  |  |  |
| Seminar paper(s) | | | 20 |  |  |

# Comments on policy proposal of introduction of short cycle programs in Republic of Serbia

Part-time studies are not jet recognized in Serbia. The low which will enable this kind of studies is in preparation still. In accordance with fact, this following concept of part-time studies in VIŠER is just a proposal, and it will be changed according to the low when the parliament of Republic of Serbia changes this part of low which concerns high education.

## SHORT CYCLE STUDY

|  |
| --- |
| Duration od study |
| Short cycle Programming consists of two modules: Web Application Programmer and Computer Programmer/Analyst. Duration of the selected module is 12 months. |

|  |
| --- |
| Number of ECTS |
| Short cycle Programming consists of 24 courses. Courses are split into 12 disjunctive courses per module (Web Application Programmer and Computer Programmer/Analyst). Each course in the short cycle module is worth 6 ECTS. The student chooses 6 out of 12 courses in the module and by doing so sums up 36 ECTS at the end. |

|  |
| --- |
| Recognition of ECTS gained on short cycle program for purposes of enrollment other study programs |
| Student who enrolls the first year of undergraduate academic studies or undergraduate applied studies may require admission of courses that are he or she has accomplished within the short cycle Programming. |

|  |
| --- |
| Quality |
| The quality aspect of the short cycle Programming is directly focused on the requirements of the IT sector. The main objective is a high student employment rate upon completion of short cycle Programming. |

|  |
| --- |
| Organizer and executor of short cycle program |
| Organizer and the bidder of short cycle Programming is the School of Electrical and Computer Engineering of Applied Studies, Belgrade, which has three IT related accredited study programs. |

|  |
| --- |
| Accreditation |
| The School of Electrical and Computer Engineering and Computing of Applied Studies, Belgrade, within three IT related accredited study programs has an quality control support which is regularly carried out during each semester. Also, all schools legal acts are in accordance with the Law on Higher Education and other laws pertaining to this type of education. |

|  |
| --- |
| National register of short cycle programs |
| The School of Electrical and Computer Engineering of Applied Studies, Belgrade, delivers information on short cycle Programming through the Commission for Accreditation and Quality Assurance, which is formed by the National Register of short programs. |

|  |
| --- |
| Organization of teaching process |
| Tuition feefor part-time studnets should be the same as for regular students in Information Systems program. Tuition fee is defined as numbero of ECTS multiply by cost of ome ECTS. The price of one ECTS is defined by VISER every for every school year. Student who choose part-time study must pay tuition fee.   1. Short cycle Programming is not is not designed for traditional teaching. 2. Short cycle Programming will be implemented as online teaching where student uses e-learning system at the School of Electrical and Computer Engineering of Applied Studies, Belgrade. Student accesses to appropriate teaching materials, while the communication with course teachers and teaching assistants is mandatory, as well as with other students. It is also possible to use other forms of teaching that are based on the usage of modern electronic and communication technologies. 3. Combined teaching in the short cycle Programming takes place in accordance with the estimation provided by professors, course topics and the requirement for equipment that student does not own at home.   Teaching on short cycle Programming is independent of the teaching on applied studies. |

|  |
| --- |
| Who can be a student of short cycle program? |
| Student of short cycle Programming can be a person with who completed a high school education, a person who has dropped out the undergraduate study programs, as well as persons who have completed the undergraduate academic studies, undergraduate applied studies, master academic studies and master applied studies. Registration and the number of students that enroll the short cycle Programming is governed by the laws of the School of Electrical and Computer Engineering of Applied Studies, Belgrade. |

|  |
| --- |
| Teaching staff |
| Short cycle Programming teaching staff includes employed teachers and teaching assistants of the School of Electrical and Computer Engineering of Applied Studies, Belgrade, or persons engaged by engagement agreement or contract on additional work that posses appropriate skills. |

|  |
| --- |
| Preparation of curriculum of short cycle program and the supporting documents |
| Documentation accompanying short cycle Programming includes:   1. Administrative data on VISER. 2. Contract with at least one employer willing to hire students upon completion of short cycle (employer reserves the right to choose employee candidates), or with an employer who is willing to send their employees to further training through the short cycle. 3. Name, goal, outcomes, the scope of the program (expressed through ECTS or otherwise). 4. Description of the specific job for which short cycle students are prepared for and the necessary competences that they must possess in order to complete the cycle. 5. The curriculum of the cycle with the teaching and other activity schedule. 6. Course books for each short cycle course, consisting of: purpose, outcomes of learning and training course contents, teaching methods, methods of evaluating the knowledge and examination, literature and number of ECTS given to students that will pass the final exam. 7. Admission requirements and registration procedures. 8. List of books of short cycle teaching staff. 9. Obligations of the student. 10. The size of teaching groups. 11. Procedures for short cycle quality management. 12. Tuition, i.e. the price that student must pay during registration that will allow a student to continue with the short cycle. |

|  |
| --- |
| Approval of short cycle program |
| Short cycle Programming is prepared in accordance with the Regulations on the procedure of short cycle preparation, approval and quality control and approved by the Academic Council of the School of Electrical and Computer Engineering of Applied Studies, Belgrade, for a period of three years in accordance with the positive internal assessment and demand in the labor market related to students who have completed short cycle Programming. Rule book includes representatives of employers interested in hiring students who complete short cycle Programming or employers who are interested in training their employees. |

|  |
| --- |
| Internship |
| Short cycle Programming includes an internship in a company which, working on a position that the student is being educated for. Duration of the internship is 10% of the duration of the short cycle. |

|  |
| --- |
| National certificate |
| A student who completes all the exams receives a national certificate on completion of the short cycle. This public document validates that the student is trained and qualified to perform a specific job. In addition to the certificate the following is given:   1. list of the courses successfully completed, with associated ECTS, the scope of the teaching process, and final grade; 2. knowledge and skills as well as learning outcomes, which are relevant to the job that the certificate is issued for; 3. job description that the certificate holder is qualified for. |

|  |
| --- |
| Partial realization of the program |
| A student can choose to attend teachings of one or more short cycle Programming subjects. In this case, the School of Electrical and Computer Engineering of Applied Studies, Belgrade, issues a certificate on passed exams to a student (certificate on the accomplished part of the cycle; final grade transcript), which gives information about the cycle (cycle content and learning outcomes) as well as student's final grades. If the School of Electrical and Computer Engineering of Applied Studies, Belgrade, offers the opportunity to enroll only in individual courses, the school must provide tuition cost for each course separately. |

|  |
| --- |
| Financing of short cycle programs |
| Short cycle Programming is financed by tuition costs paid by students or employers. In specific cases, when the government wants to encourage short cycle studies to educate students for new jobs or positions that lack adequate staff on the labor market, fees, or part thereof, for such students may be paid by the competent state authority or the responsible organization. In this case, the authority or organization shall publish a tender for the participant in the cycles which want to give financial support for. One of the participants is the School of Electrical and Computer Engineering of Applied Studies. |