PT&SCHE
Introduction of part-time and short cycle studies in Serbia

Co-funded by the Erasmus+ Programme of the European Union

EVALUATION OF THE PILOT SHORT CYCLE PROGRAM "PROGRAMMING IN JAVA" AT BELGRADE METROPOLITAN UNIVERSITY

Dragan Domazet[1]

E-mail: dragan.domazet@metropolitan.ac.rs

Jovana Jović[2]

E-mail: jovana.jovic@metropolitan.ac.rs

Abstract: *This paper presents an evaluation of the short program "Programming in Java" implemented at Belgrade Metropolitan University (BMU) as a pilot program of the project "Introduction of part-time and short cycle studies in Serbia" (PT&SCHE), funded by Erasmus+ European Union program realized from October 2015 to May 2018. Based on the analysis of the most appropriate model for the adoption and development of PT&SCHE in Serbia, this paper aims to analyze and discover the most relevant pedagogical and methodological approach in eLearning processes of short programs. This paper describes the teaching methodology and e-learning technology proposed and implemented by BMU for short cycle programs (SCHE). The paper also analyzes the effects to students (their success and satisfaction), it provides evaluation of the applied methodology in the pilot phase of the short program "Programming in Java" realized by BMU.*

KEY WORDS: E-LEARNING, SHORT CYCLE PROGRAM, PT&SCHE PROJECT

1. INTRODUCTION

The Higher Education Act (2017), provided legal bases for part-time studies (PT) and short cycle programs in higher education (SCHE). Belgrade Metropolitan University (BMU) got the assignment to develop and evaluate a short cycle program according to the new Higher Education Act and recommendations generated by the PT&SCHE project, as pilot implementation of a SCHE or a "short program", as specified by the Higher Education Act. "

A short program (SCHE) enables a student to become capable of performing a particular job (for example, a Java programmer), as opposed to 3 or 4 year study programs that educate a student for a specific profession in a particular field (e.g. Information Technology Engineer). In this sense, the program is a set of courses that provides students with qualifications for a particular job within 3 to 12 months. While study programs of bachelor and master studies in a period of three or four years train a student for jobs in one profession (for example, software engineering), a short program enables a student ro do a particular job within a profession (eg, Java programmer) within 12 months A short program can provide a qualification for a particular job, within a profession, at the 5th, 6th or 7th level EQF (2008), depending on the job complexity and requirements.

After an analysis of the current labour market needs, BMU decided to develop a pilot SCHE program named "Programming in Java" with the aim to develop and test the most suitable methodology for

[1] Institution/Affiliation: Belgrade Metropolitan University/ Rector and Emeritus Professor

[2] Institution/Affiliation: Belgrade Metropolitan University/Research Assistant

SCHE programs. The SCHE program "Programming in Java" was developed according to the standard internal review of BMU. In addition to the carefully prepared curriculum, lecturers and tutors were carefully selected from faculty members of BMU. This work represents evaluation of the applied methodology of the whole eLearning process during the pilot phase of the realization of the program at BMU with the aim to define in what extent the proposed methodology was successful and what needs to be changed in the future.

This paper is organized as follows. Section 2 describes technical details of the curriculum and training details of the pilot online program "Programming in Java". Section 3 describes the teaching methodology applied in the short cycle program. Section 4 shows the evaluation of the applied methodology. Section 5 concludes the paper.

2. THE CURRICULUM

2.1 Specification of the job profile "Java Developer"

"European ICT Professional Profiles" (2017) specifies a European ICT Profiles family tree structured from six main ICT Profile families. (Figure 1.). "The concept devised is broadly analogous to human genetics where the genes of one generation pass down to the next. In the same way it is envisaged that the core components of the 23 Generation 2 Profiles will pass down to profiles constructed to meet specific stakeholder requirements. The 23 Profiles constructed in this CWA combined with e-competences from the e-CF (2014), provide a gene pool for the development of tailored profiles that may be developed by European ICT sector players in specific contexts and with higher levels of granularity.

The 23 multi-stakeholders agreed that ICT Profile descriptions are based on the European e-Competence Framework (e-CF). European ICT Profiles and e-Competence are complementary concepts that can significantly support the development and management of a world class ICT professional community within Europe.

Applied at the same level of granularity as the e-CF, the European ICT Profiles provide generic skeletons of the most representative Profile prototypes currently used in ICT Business structures." Figure 2 shows the job profile "Developer" as specified in "European ICT Professional Profiles" (2017).

The European e-Competence Framework (e-CF) version 3.0 provides a reference of 40 competences as required and applied at the Information and Communication Technology (ICT) workplace, using a common language for competences, skills and capability levels that can be understood across Europe. For the Developer position, it has been defined that it must have five e-competencies (listed in the E-Competence Framework document (e-CF)):

B.1. Design and Development (Level 3)

B.2. System Integration (Level 2)

B.3.Testing (Level 2)

B.5. Documentation Production (Level 3)

C.4. Problem Management (Level 3)

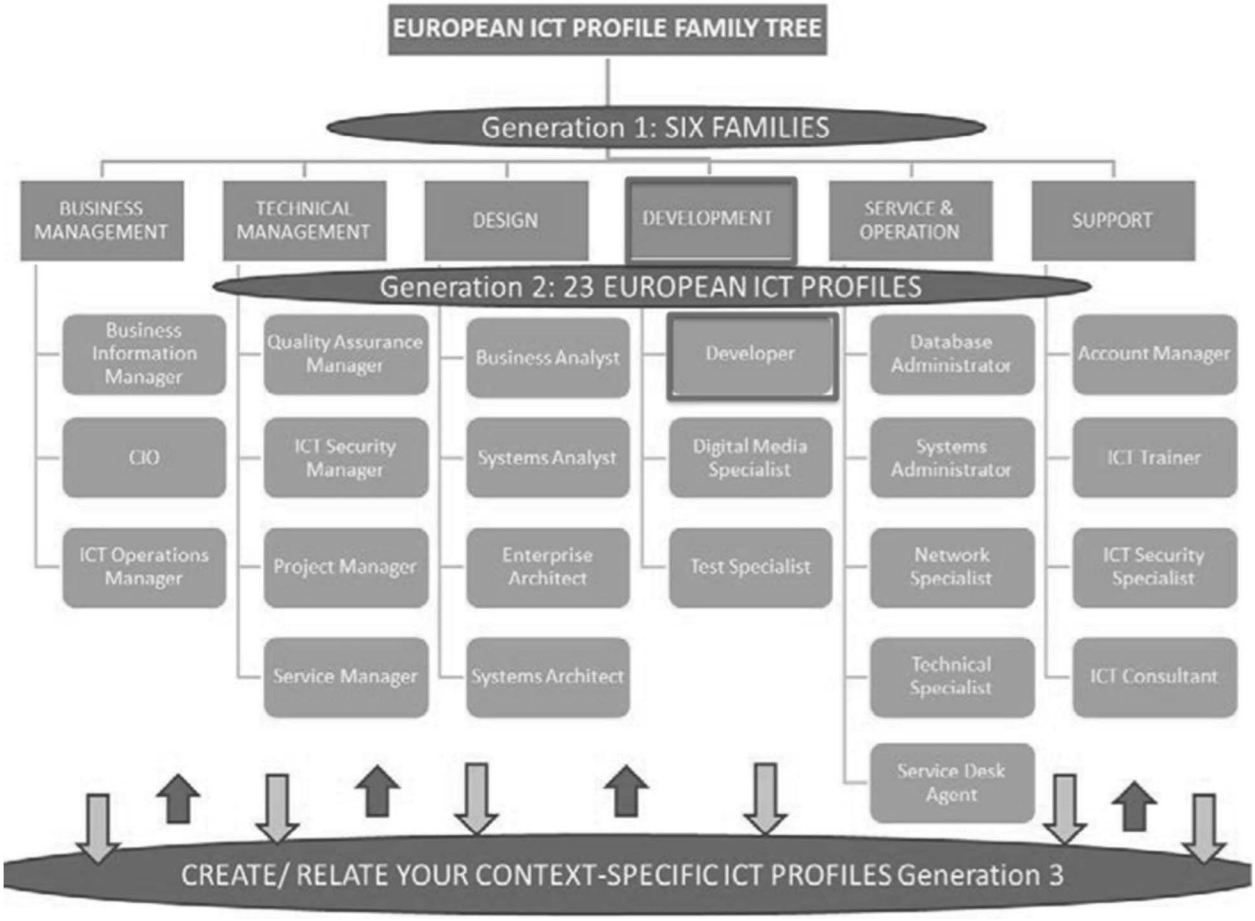


Figure 1.

To illustrate one of the specified five e-competences, we present the e-competence B.1. Design and Development in Figure 3.

Depending on the achieved e-competence level (e-2 or e-3) and EQF level (5 or 6), an SCHE program may educate and train a Java Junior Developer or a Java Developer (Figure 4). The pilot implementation od the SCHE Programming in Java is developed for Java Developer level (e-3 and EQF level 6).

Profile title	**DEVELOPER** (6)		
Summary statement	Builds/codes ICT solutions and specifies ICT products according to customer needs.		
Mission	Ensures building and implementing of ICT applications. Contributes to planning, low level design. Compiles diagnostic programs and designs and writes code for operating systems and software to ensure optimum efficiency and functionality.		
Deliverables	**Accountable**	**Responsible**	**Contributor**
	• Hardware Component • Software Component	• Solution Documentation	• Software Design Description • Test Procedure • Solution in Operation
Main task/s	• Develop component • Engineer component • Shape documentation • Provide component support beyond the first level • Supply 3rd level support		
e-competences *(from e-CF)*	B.1. Design and Development		Level 3
	B.2. Systems Integration		Level 2
	B.3. Testing		Level 2
	B.5. Documentation Production		Level 3
	C.4. Problem Management		Level 3
KPI area	Fully functional ICT components		

Figure 2.

An elaboration of the e-competences B.1, B.2, B3., B.5 and C.4 of a Developer are given in the following tables, based on their specification given in e-CF (2014). We specify two competence levels of Java developers: **JUNIOR JAVA DEVELOPER** (e-2) and **JAVA DEVELOPER** (e-3)

Table 1: Job description related to different e-competences

	JUNIOR JAVA DEVELOPER & JAVA DEVELOPER
e-competences	**Job Description**
B.1. Application Development	Interprets the application design to develop a suitable application in accordance with customer needs. Adapts existing solutions by e.g. porting an application to another operating system. Codes, debugs, tests and documents and communicates product development stages. Selects appropriate technical options for development such as reusing, improving or reconfiguration of existing components. Optimises efficiency, cost and quality. Validates results with user representatives, integrates and commissions the overall solution.
B.2. Component Integration	Integrates hardware, software or sub system components into an existing or a new system. Complies with established processes and procedures such as, configuration management and package maintenance. Takes into account the compatibility of existing and new modules to ensure system integrity, system interoperability and information security. Verifies and tests system capacity and performance and documentation of successful integration.
B.3.Testing	Constructs and executes systematic test procedures for ICT systems or customer usability requirements to establish compliance with design specifications. Ensures that new or revised components or systems perform to expectation. Ensures meeting of internal, external, national and international standards; including health and safety, usability, performance, reliability or compatibility. Produces documents and reports to evidence certification requirements.
B.5. Documentation Production	Produces documents describing products, services, components or applications to establish compliance with relevant documentation requirements. Selects appropriate style and media for presentation materials. Creates templates for document-management systems. Ensures that functions and features are documented in an appropriate way. Ensures that existing documents are valid and up to date.
C.4. Problem Management	Identifies and resolves the root cause of incidents. Takes a proactive approach to avoidance or identification of root cause of ICT problems. Deploys a knowledge system based on recurrence of common errors. Resolves or escalates incidents. Optimises system or component performance.

Table 2: e -Competence levels

	JUNIOR JAVA DEVELOPER	JAVA DEVELOPER
e-competences	**Level e-2**	**Level e-3**
B.1. Application Development	Systematically develops and validates applications.	Acts creatively to develop applications and to select appropriate technical options. Accounts for others development activities. Optimizes application development, maintenance and performance by employing design patterns and by reusing proved solutions.
B.2. Component Integration	Acts systematically to identify compatibility of software and hardware specifications. Documents all activities during installation and records deviations and remedial activities.	As for Level e-2
B.3.Testing	Organises test programmes and builds scripts to stress test potential vulnerabilities. Records and reports outcomes providing analysis of results.	As for Level e-2
B.5. Documentation Production	Determines documentation requirements taking into account the purpose and environment to which it applies.	Adapts the level of detail according to the objective of the documentation and the targeted population.
C.4. Problem Management	Identifies and classifies incident types and service interruptions. Records incidents cataloguing them by symptom and resolution.	Exploits specialist knowledge and in-depth understanding of the ICT infrastructure and problem management process to identify failures and resolve with minimum outage. Makes sound decisions in emotionally charged environments on appropriate action required to minimise business impact. Rapidly identifies failing component, selects alternatives such as repair, replace or reconfigure.

Table 3: Knowledge needed related to different e-competences

	JUNIOR JAVA DEVELOPER & JAVA DEVELOPER
e-competences	**KNOWLEDGE: Knows/aware of/ familiar with**
B.1. Application Development	K1 appropriate software programs/modules K2 hardware components, tools and hardware architectures K3 functional & technical designing K4 state of the art technologies K5 programming languages K6 power consumption models of software and/or hardware K7 DBMS K8 operating systems and software platforms K9 Integrated development environment (IDE) K10 rapid application development (RAD) K11 IPR issues K12 modelling technology and languages K13 interface definition languages (IDL) K14 security
B.2. Component Integration	K1 old, existing and new hardware components/software programs/modules K2 the impact that system integration has on existing system/organisation K3 interfacing techniques between modules, systems and components K4 integration testing techniques K5 development tools (e.g. development environment, management, source code access / revision control) K6 best practice design techniques
B.3.Testing	K1 techniques, infrastructure and tools to be used in the testing process K2 the lifecycle of a testing process K3 the different sorts of tests (functional, integration, performance, usability, stress etc.) K4 national and international standards defining quality criteria for testing K5 web, cloud and mobile technologies and environmental requirements
B.5. Documentation Production	K1 tools for production, editing and distribution of professional documents K2 tools for multimedia presentation creation K3 different technical documents required for designing, developing and deploying products, applications and services K4 version control of documentation production
C.4. Problem Management	K1 the organisation's overall ICT infrastructure and key components K2 the organisation's reporting procedures K3 the organisation's critical situation escalation procedures K4 the application and availability of diagnostic tools K5 the link between system infrastructure elements and impact of failure on related business processes.

Table 4: Skills needed related to different e-competences

	JUNIOR JAVA DEVELOPER & JAVA DEVELOPER
e-competences	**SKILLS: is able to**
B.1. Application Development	S1 explain and communicate the design/development to the customer S2 perform and evaluate test results against product specifications S3 apply appropriate software and/or hardware architectures S4 develop user interfaces, business software components and embedded software components S5 manage and guarantee high levels of cohesion and quality S6 use data models S7 perform and evaluate test in the customer or target environment S8 cooperate with development team and with application designers
B.2. Component Integration	S1 measure system performance before, during and after system integration S2 document and record activities, problems and related repair activities S3 match customers' needs with existing products S4 verify that integrated systems capabilities and efficiency match specifications S5 secure/back-up data to ensure integrity during system integration
B.3.Testing	S1 create and manage a test plan S2 manage and evaluate the test process S3 design tests of ICT systems S4 prepare and conduct tests of ICT systems S5 report and document tests and results
B.5. Documentation Production	S1 observe and deploy effective use of corporate standards for publications S2 prepare templates for shared publications S3 organise and control content management workflow S4 keep publications aligned to the solution during the entire lifecycle
C.4. Problem Management	S1 monitor progress of issues throughout lifecycle and communicate effectively S2 identify potential critical component failures and take action to mitigate effects of failure S3 conduct risk management audits and act to minimise exposures S4 allocate appropriate resources to maintenance activities, balancing cost and risk S5 communicate at all levels to ensure appropriate resources are deployed internally or externally to minimise outages

All specified competences, related to job description, levels of e-competences, knowledge needs and skills needs are implemented in Java. Due to time limitation specified by Higher Education Low (2017) of only 12 months, and having in mind the complexity of Java technology and its implementation, BMU decided to develop an SCHE program with the aim to train future **Junior Java Developer.** After appropriate experience, they can enroll to another SCHE program aiming to train future Java Developers.

Based on above, we specified Generation 3 of the ICT Profile Family, as shown in Figure 3. It describes two job profiles Junior Java Developer (e-2, EQF Level 5) and Java Developer (e-3, EQF Level 6). Specifications for these Generation 3 job profiles are given in Tables 1-4. The major difference between Generation 2 Developer and Generation 3 Java developer profiles is in the area of development. Java Developer uses Java technology to develop an application. The pilot implementation of our short (SCHE) program was developed and implemented for the job profile **Junior Java Developer.**

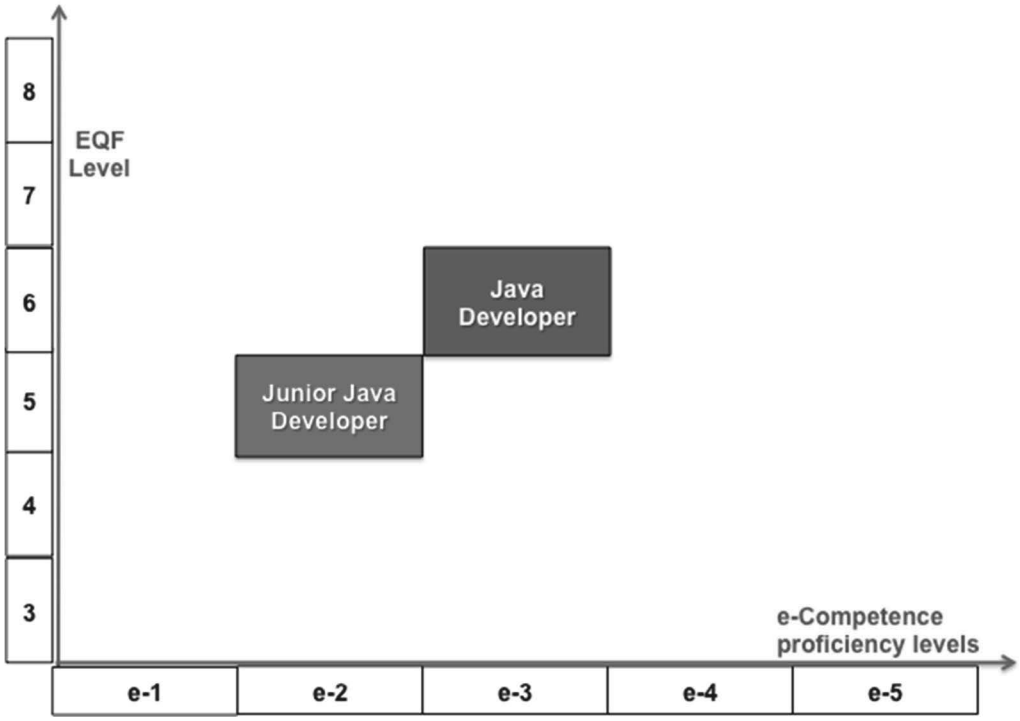


Figure 3.

In order to distinguish these two Generation 3 job profiles, we cite description elements of EQF Levels and e-CF Levels in Table 5 using the relevant part of the table given in e-CF(2014).

Table 5: European e-CF and EQF level table

EQF Level	EQF Levels descriptions	e-CF Levels	e-CF Levels descriptions	Typical Tasks	Complexity	Autonomy	Behaviour
6	Advanced knowledge of a field of work or study, involving a critical understanding of theories and principles, advanced skills, demonstrating mastery and innovation in solving complex and unpredictable problems in a specialised field of work or study, management of complex technical or professional activities or projects, taking responsibility for decision-making in unpredictable work or study contexts, for continuing personal and group professional development.	e-3	**Senior Professional/Manager** Respected for innovative methods and use of initiative in specific technical or business areas; providing leadership and taking responsibility for team performances and development in unpredictable environments.	Consulting	Structured – unpredictable	Works independently to resolve interactive problems and addresses complex issues. Has a positive effect on team performance.	Planning, making decisions, supervising, building teams, forming people, reviewing performances, finding creative solutions by application of specific technical or business knowledge / skills.
5	Comprehensive, specialised, factual and theoretical knowledge within a field of work or study and an awareness of the boundaries of that knowledge, expertise in a comprehensive range of cognitive and practical skills in developing creative solutions to abstract problems, management and supervision in contexts where there is unpredictable change, reviewing and developing performance of self and others.	e-2	**Professional** Operates with capability and independence in specified boundaries and may supervise others in this environment; conceptual and abstract model building using creative thinking; uses theoretical knowledge and practical skills to solve complex problems within a predictable and sometimes unpredictable context.	Concepts / Basic principles	Structured – unpredictable	Works under general guidance in an environment where unpredictable change occurs. Independently resolves interactive issues which arise from project activities.	Designing, managing, surveying, monitoring, evaluating, improving, finding non standard solutions.

2.2 The Body of Knowledge

The specification of the job profiles according to the European ICT Professional Profiles (2017) and European e-Competence Framework (e-CF) was a good guideline for us to describe the job profile that our short program (SCHE) has to provide. It specifies key e-competences with four dimensions and five possible levels (dimension 3, as shown in Figures 3 and 5. Dimension 4 specifies key knowledge areas and skills of a job profile, for e-competences B1-B5 and C4. These competences are given in Domazet (2017).

Competences specified for Java Developer profile, are not sufficient to specify the curriculum of the SCHE program "Programming in Java", as more detailed knowledge specification is needed. When a curriculum has to be developed, it is usually necessary to develop its Body of Knowledge (BOK). We developed BOK for SCHE program "Programming in Java" based on the following relevant and existing BOKs:

- The Foundation ICT BOK (2015)
- SWEBOK 3.0 (2014)
- Computer Science BOK (2013)

The Foundation ICT BOK (2015) specifies 12 knowledge areas:

1. ICT Strategy & Governance
2. Business and Market of ICT
3. Project Management
4. Security Management
5. Quality Management
6. Architecture
7. Data and Information Management
8. Network and Systems Integration
9. Software Design and Development
10. Human Computer Interaction
11. Testing
12. Operations and Service Management.

Four of these 12 knowledge areas are relevant for SCHE program "Programming in Java":

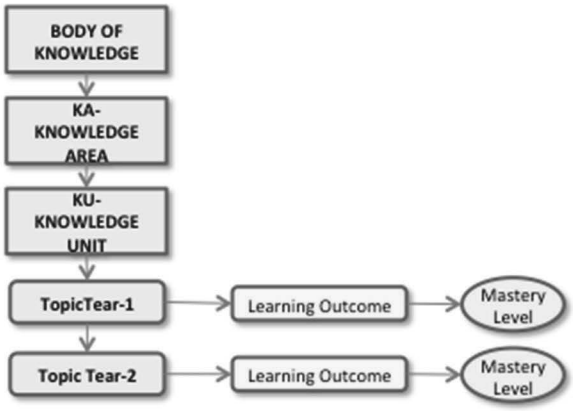
- Software Design and Development
- Human Computer Interaction
- Data and Information Management
- Testing

All four knowledge areas are presented in Domazet (2017).

Unfortunately, the ICT Foundation Body of Knowledge does not provide yet lower levels of knowledge and it is not sufficient for curriculum development. Therefore, additional extensions (sub-topics) of the Bodies of Knowledge are needed. We used two BOKs: SWEBoK 3.0 (2014) and Computer Science BOK (2013), shown in Figure 4. BMU is using these two BOKs for its BSc programs: Software Engineering and Information Technology. The development of our SCHE courses are therefore based on these two BOKs, and courses of our two BSc degree programs: Software Engineering and Information Technologies (Figure 5)

Computer Science BOK 2013

No.	Knowledge Area	Level 1	Level 2
1	• AL - Algorithms and Complexity	19	9
2	• AR - Architecture and Organization	0	16
3	• CN - Computational Science	1	0
4	• DS - Discrete Structures	37	4
5	• GV - Graphics and Visual Computing	2	1
6	• HCI - Human-Computer Interaction	4	4
7	• IAS - Information Assurance and Security	2	6
8	• IM - Information Management	1	9
9	• IS - Intelligent Systems	0	10
10	• NC - Networking and Communications	3	7
1	• OS - Operating Systems	4	11
12	• PBD - Platform-based Development	0	0
13	• PD - Parallel and Distributed Computing	5	10
14	• PL - Programming Languages	8	20
15	• SDF - Software Development Fundamentals	43	0
16	• SE - Software Engineering	6	21
17	• SF - Systems Fundamentals	18	9
18	• SP - Social Issues and Professional Issues	11	5
	Total hours	164	142



SWEBOK-V3 (2014)

RB	Akronym	Naziv oblasti znanja (KNOWLEDGE AREA)
1	SR	SOFTWARE REQUIREMENTS
2	SD	SOFTWARE DESIGN
3	SC	SOFTWARE CONSTRUCTION
4	ST	SOFTWARE TESTING
5	SM	SOFTWARE MAINTENANCE
6	SCM	SOFTWARE CONFIGURATION MANAGEMENT
7	SEM	SOFTWARE ENGINEERING MANAGEMENT
8	SEP	SOFTWARE ENGINEERING PROCESS
9	SEMM	SOFTWARE ENGINEERING MODELS AND METHODS
10	SQ	SOFTWARE QUALITY
11	SEPP	SOFTWARE ENGINEERING PROFESSIONAL PRACTICE
12	SEE	SOFTWARE ENGINEERING ECONOMICS
13	CF	COMPUTING FOUNDATIONS
14	MF	METHEMATICAL FOUNDATIONS
15	EF	ENGINEERING FOUNDATIONS

Figure 4.

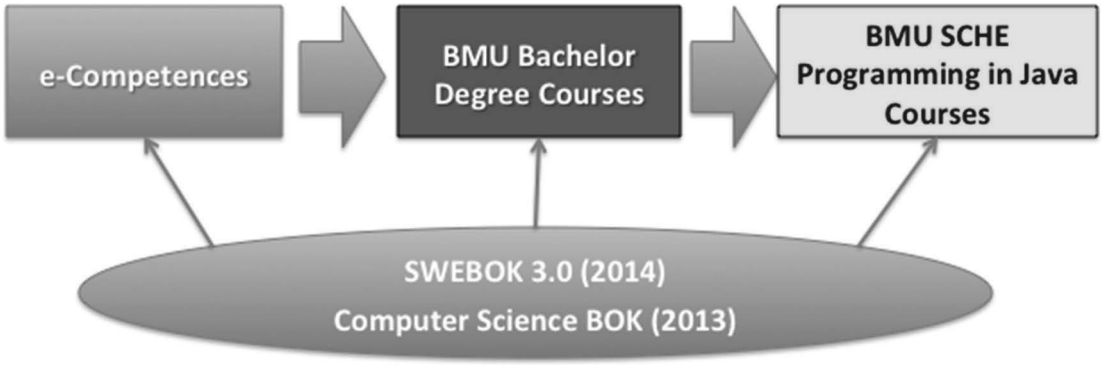


Figure 5.

The curriculum of the SCHE program “Programming in Java” is determined by mapping of relevant parts (learning units) of the BMU Bachelor's degree programs. It is implemented in three stages where each of them has one or more courses:

1. **Preparatory stage** – aiming to prepare trainees for programming training, providing the some basic knowledge in IT systems and programming fundamentals. The following courses are included:
   1.1. KI101 Introduction to IT systems
   1.2. KI102 Fundamentals of Programming
2. **Learning stage** – providing programming knowledge and skills to trainees, as well as some basic soft skills that might be useful for their employability. This stage includes:
   2.1. KI103 Java 1: Fundamentals of Programming
   2.2. KI104 Java 2: Object-oriented Programming
   2.3. KI105 Java 3: GUI Programming
   2.4. KI201 Java 4: Data Structures and Algorithms - Part A
   2.5. KI202 Java 5: Data Structures and Algorithms - Part B
   2.6. KI203 Java 6: Advanced Java Programming
   2.7. KI204 Java 7: Java Enterprise Edition
   2.8. KI205 Java 8: Java Programming on the Android platform
   2.9. KI206 Software Development Process and Methodology
   2.10. KI301 Software Construction
3. **On-the-job training stage** – providing trainees with one course (KI401) and one internship (KI402):
   3.1. KI401 Software Development Project
   3.2. KI402 Professional Internship - Java Developer

Table 6 shows core courses with their course hours of all listed courses and dates of their start.

In creating a short program, several IT firms, especially those dealing with software development using Java technology, have been consulted to ensure that this short program is created by "tailor-made employers" who need to hire students who complete this program. The final quality indicator of the realized short program is the percentage of student employment in the first three months after the completion of the short program. For each of the above defined courses, the program contents and learning outcomes are defined, so in the end, they provide the required competencies for the Java programmer work profile.

The program provides 600 hours of active teaching and 60 ESPB, i.e. credits that can be recognized if the student decides to enroll later on one of four BSC degree programs of BMU: Software Engineering, Information Technology, Computer Games and Information Systems. In the normal duration of the 12 month program, students will have: nine months for active learning, two months of internship and one month for annual leave. Learning is performed in blocks, i.e. according to the "course-by-course" system. It is planned that on each course, a student spends at least three hours a day. Including Saturdays (or 18 hours per week) using the e-Learning System of BMU. Table 8 shows start- and end-days of major groups of training activities of the SCHE program “Programming in Java”, lasting, in total, 12 months and providing 60 ECTS.

Syllabi of BMU SCHE program “Programming in Java” for all courses of the program are specified in Domazet (2017). They are implementing parts of Computer Science BOK (2013) and SWEBOK (2014).

Table 6: Core courses of the SCHE program "Programming in Java"

Short-Cycle Programme for Junior Java Developer

Duration: 12 months 60 ECTS

#	Course	Duration (Days)	Teaching Days	Workshop Days	Teaching Hours	ECTS	Starting Date
1	KI101 Introduction to IT systems	15	14	0	42	4	02/10/2017
2	KI102 Programming Fundamentals	11	8	2	30	3	23/10/2017
3	KI103 Java 1: Fundamentals of Programming	17	14	2	48	5	06/11/2017
4	KI104 Java 2: Object-oriented programming	13	10	2	36	3	27/11/2017
5	KI105 Java 3: GUI Programming	17	14	2	48	4	11/12/2018
6	KI201 Java 4: Data Structures and Algorithms – Part A	16	13	2	45	4	08/01/2018
7	KI202 Java 5: Data Structures and Algorithms – Part B	16	13	2	45	4	29/01/2018
8	KI203 Java 6: Advanced Java Programming	15	12	2	42	4	19/02/2018
9	KI204 Java 7: Java Enterprise Edition	24	21	2	69	7	12/03/2018
10	KI205 Java 8: Pragramming in Java on Android Plaform	14	11	2	39	4	09/04/2018
11	KI206 Development Process and Methodologies	18	15	2	51	5	30/04/2018
12	KI301 Software Construction	21	18	2	60	6	21/05/2018
13	KI401 Software Development Project	6	5	10	45	4	18/06/2018
14	KI402 Intership (8 weeks)	40	0	0	0	3	06/08/2018
	TOTAL:	**243.0**	**168.0**	**32.0**	**600.0**	**60.0**	

Table 7. Time schedule of courses

	Start Date	End Date
Introdutory courses	1.10.2017	5.11.2017
Online training courses	7.11.2017	13.7.2018
Summer Holidays	15.7.2018	5.8.2018
Internship (8 weeks)*	8.8.2018	28.9.2018

* Student has to realize its 8 weeks intership in this perod

The learning material provided on the BMU's e-Learning System contains web pages with multimedia contents (text, figures, Java codes, video and audio content), but also a PDF version of each lesson, the textual part of each lesson in PDF format (as an additional teaching material), and the student is not required no additional literature (although it is always useful to use other sources of knowledge). In addition to online classes, a two-day workshop in the computer classrooms of the University is planned for each course. All students get work assignments (as homework tasks) and one project per course. If a student is prevented from participation in workshops, their workshops can also be organized online. After the completed project and all assignments and tests are evaluated, students take the exam of each course.

After successfully completing professional internship in every course, the student receives the Certificate of Successful Completion of the Short Program, with an appendix containing the obtained grades on all subjects of the Short Program with a list of course learning outcomes and competences.

3. TEACHING METHODOLOGY

Figure 6 shows the organization of the training program and online learning materials. The Training Program consists of Courses. Each Course has a different number of lessons. A lesson describes the objectives, procedures, materials and evaluation for a particular class or a particular day. Each Lesson consists of one or more Learning Units. A Learning Unit provides a new and short concept depending on the content. Each Learning Unit has a clear learning objective regarding to change the level if knowledge and skills of trainees. A Learning Units consists of one or more Topics. A Topic or Sub-Topic is an atomic learning concept with a clear learning outcome.

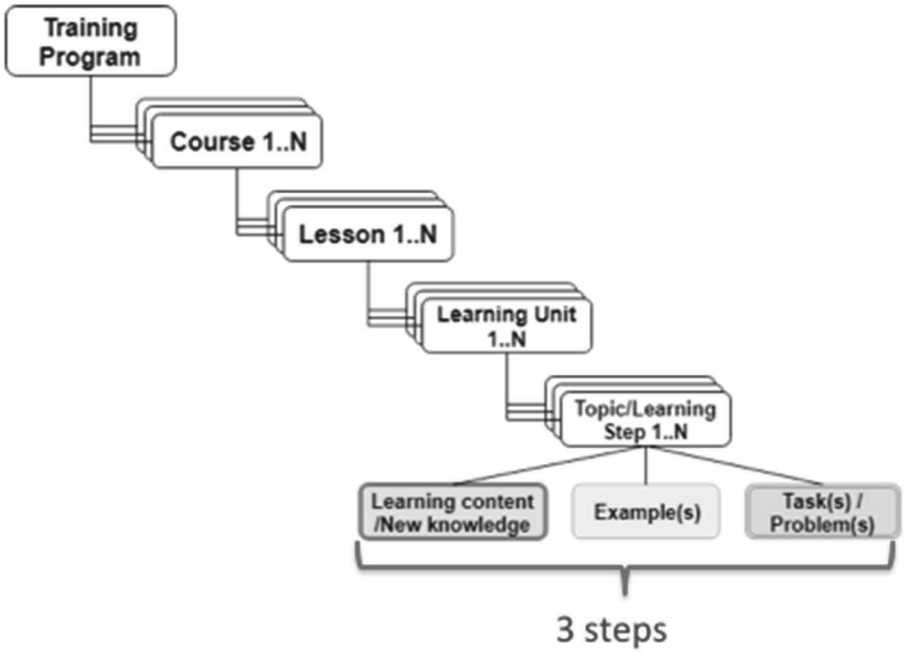


Figure 6

Especially for SCHE programs, BMU decided to implement a "Step-By-Step" learning methodology. Each Step provides small chunks of new knowledge to trainees, related to a Topic, immediately followed by one or more given examples (solved problems) and by tasks or (unsolved) problems/exercises that trainees have to solve by using this newly acquired knowledge. Figure 7 shows Learning Units with their Thematic Steps (Topics), each with three types of sub-steps:

1. New knowledge acquisition (a learning concept) – a small chunk of new knowledge or a new concept
2. Presentation of examples of solved problems demonstrating use of new knowledge
3. Tasks (unsolved problems) to be accomplished by each learners – given problems to solve

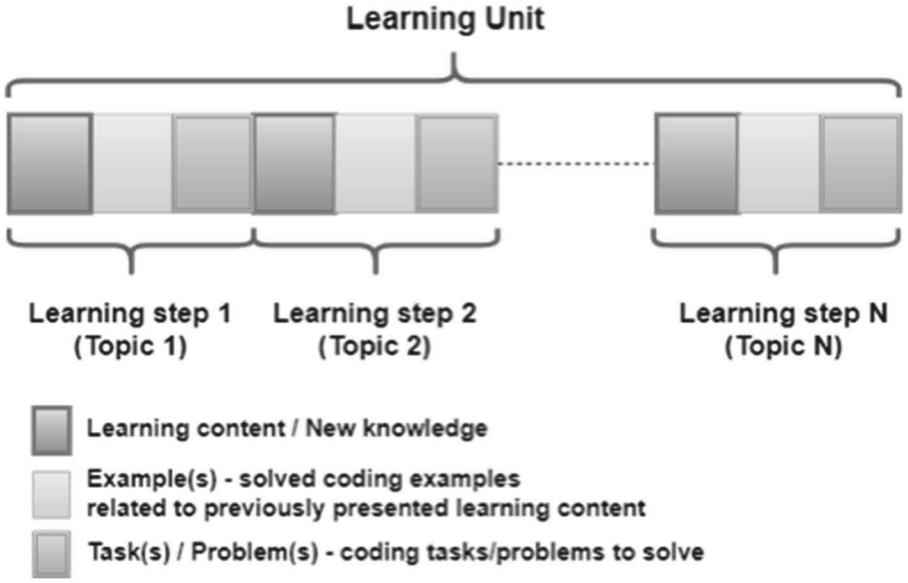


Figure 7.

The "Step-By-Step" learning methodology has been developed by BMU for SCHE programs aiming to provide high level of applied knowledge (Bloom Level 3). A trainee learns how to implement newly acquired knowledge, as a trainee must demonstrate his implementation capability by solving given problems. The granularity of Topics/Learning Steps should be as small as feasible, in order to achieve high level of integration of knowledge acquisition and its application activities. A Learning Path of a trainee is a Learning Process leading to the Learning Goal (achieving an appropriate knowledge and capability level) consisting of many Learning Activities related to hierarchically structured knowledge (Learning Units/Topics/ Sub-Topic).

BMU has structured its learning materials according to the requirement of the "Step-By-Step" methodology. BMU has a large repository of Learning Objects (LOs) ("any entity, digital or non-digital, that may be used for learning, education or training"– one for each Topic or Sub-Topic. Use of Learning Objects (LO) with fine granularity, allows easy configuration and generation of new learning material specifically created for a specific Study or Training Program. The use of LOs with fine granularity provides also a high reusability of the existing and previously developed LOs. An appropriate Learning Material may be efficiently developed for a new curriculum with high degree of reusable learning objects. By implementation the concept of LOs, with fine granularity, BMU is able to create in the learning material for "Programming in Java" training program and its curriculum in a few weeks. The online delivery of the learning material by using the BMU eLearning System, provides an efficient mechanism for the delivery of the learning materials to all trainees, and their additional interaction using its interactive activities.

4. EVALUATION

During the pilot phase, through all courses, learning activities were logged in the eLearning system (LAMS v3.0) including: time spent on learning online materials, the number of solved tasks and assessments, as well as other interactions with the resources provided by the eLearning system. The information system of BMU (ISUM) tracks all marks and other data about progress of students by course, such as: number of points per tests and projects, the number of points scored in the exam and the final grade. 8 trainees are included who passed the entire training process are included in the process of evaluation (Table 9).

Five trainees dropped out at the beginning of the course. Their reasons for dropping out are of personal nature, so we will not take them into account during the analysis. It is important to note that at the time this paper was written (September 2018), only two students successfully passed all the exams in the first test period, and six more have to pass the correction exams. In order to analyze the relatively poor results in the first test period (only 25% successfully passed all exams), we analyzed the use of learning materials and other activities provided by the eLearning system in order to determine the extent to which the system was used and what resources were most used.

Comparing the time spent in learning and number of failed exams, it is evident that trainees T3 and T8 spent less time in learning from others who have fewer failed exams. In this part, it is very important to motivate students to use the e-learning system to a greater extent somehow.

Table 9. Time spent in e-learning by trainees

Trainee	Number of failed exams (max 13)	Projected learning time in hours	Time spent in learning
T1	0	606	780
T2	3	606	585
T3	8	606	120
T4	0	606	980
T5	2	606	780
T6	3	606	330
T7	3	606	300
T8	8	606	80

At the end of the evaluation, trainees were asked to complete a questionnaire shown in Table 10. Prior to the questionnaire, the participant had to mark the place where the workshops were held (Nis or Belgrade). We identified 5 trainees from Belgrade and three trainees from Nis.

Table 10: Trainee satisfaction questionnaire

QUESTIONS	MEDIAN ±STD
Q1: How satisfied are you with the work of the instructor during the duration of the course?	4 ± 1.05
Q2: How satisfied are you with the realization of the course?	4 ± 1.09
Q3: How satisfied are you with the level of knowledge that you acquired after completing the course?	4 ± 1.11
Q4: How satisfied are you with the quality of teaching materials?	4 ± 0.99
Q5: How satisfied are you with the organization of teaching materials by the "step by step" rule?	4 ± 1.09
Q6: How much is the weight of the teaching material adequately distributed?	3.5 ± 0.7
Q7: How satisfied are you with the LAMS system?	5 ± 0.99
Q8: How satisfied are you with the use of the LAMS system in the form of tests and other learning activities?	5 ± 0.48
Q9: To what extent are practical examples relevant to the material?	4 ± 0.83
Q10: To what extent is it necessary to add materials to external resources?	3.5 ± 0.83
Q11: If it is necessary to add resources to the teaching materials that would be?	/
Q12: Assess the adequacy of teaching materials in line with the latest IT developments	3.5 ± 0.99
Q13: Assess the quality of the applicability of the acquired knowledge after the course passed?	3 ± 1.11
Q14: How satisfied are you with the quality of the skype consultation?	4 ± 0.93
Q15: How satisfied are you with the quality of mail consultation?	4 ± 1.16
Q16: How satisfied are you with the quality of the workshops held during the course?	5 ± 0.7
Q17: How satisfied are you with the number of classes scheduled for the course during the course?	3 ± 0.86
Q18: Choose the model that you find most suitable for the realization of workshops (once a week, once in two weeks, half and at the end of the course, at the end of the course, other)	/
Q20: How satisfied with the compatibility of exams tasks and teaching material	5 ± 0.7
Q21: Choose the model that you find most appropriate for the course that you attended (a longer course, a less intensive course or a more intensive course, a shorter period of time)	/
Q22: How satisfied are you with the support of the services of Belgrade Metropolitan University?	4 ± 1.4
Q23: Which areas in this course do you consider necessary to be improved further:	/
Q24: Here you can write your suggestions, positive feedback or possible comments about the course	/

The questionnaire used the five-point Likert scale, ranging from the lowest point (1) to the highest (5), and additional essay questions. The questionnaire also provided an option to enter comments in order to allow students to give their suggestions and comments in the form of an open-ended question. Questions Q1-3, Q14-17 and Q22 had the goal to give answers to the questions about the satisfaction of the course, instructors, organization and support from BMU services during the course. Analyzing those results students expressed that are satisfied with instructors (Q1) 4 ± 1.05 (with the median 4.00) and with realization (Q2) and level of knowledge (Q3) that are acquired after completing the course (respectively 4 ± 1.09 and 4 ± 1.11 with the median 4.0). Also, the participants answered that they are satisfied with the method of carrying out Skype consultations (Q14), 4 ± 0.93 (with the median 4.0) and a little less mail

consultation (Q15) 4 ± 1.16 (with the median 4.0), which is logical in relation to the type of course where it is sometimes difficult to explain the problem in writing. Trainees consider the workshops as very useful especially that they were organized in the traditional way in classrooms (Q16), 5 ± 0.7 (with the median 5.0), but consider that there is insufficient number of traditional classes of workshops that are necessary for an adequate understanding of the material (Q17), 3 ± 0.86 (with the median 3.0).

Some questions have a relatively high standard deviation due to the limited number of trainees, but at this stage we can conclude that trainees are satisfied with quality (4 ± methodology (4 ± 1.09) of learning materials thru Q4 and Q5, but also the trainees opinion is that the difficulty of the teaching material is not adequately distributed, 3.5 ± 0.7 (with the median 3.0), although they think that the examples (Q9) and exams (Q20) are relevant to teaching material 4 ± 0.83 (with the median 4.0). Trainees are absolutely satisfied with the BMU eLearning system (Q8) 5 ± 0.99 and they agreed that tests and other learning activities are of the great help in the learning process (Q9) 5 ± 0.48 (both with the median 5.0).

Lower satisfaction of the students is observed in the assessment of the conformity of learning materials with IT trends (Q12) 3.5 ± 0.99 and the applicability of the acquired knowledge after the course (Q13) 3 ± 1.11. We believe that such results may be due to insufficient practice of the participants and overloading information that needs to be channeled through practical work. Certainly, BMU has the task of further improving the teaching materials in order to increase the adequacy.

Students responded positively to “Programming in Java” short cycle program even through their comments through Q11, Q18, Q221, Q23 and Q24. Students considered the course to be useful and interesting:

- “Program is good”,
- “It's a good and interesting course”,
- “The program is very useful and provides a lot of practical knowledge for future employment.”

Additional comments and suggestions point to other questions of an essay questionnaire are generally similar and we can conclude that it is necessary to improve:

- “Everything is good, but there is a lack of traditional work in the classroom with instructors”,
- “More teamwork is needed”,
- “The course needs to be slower and less intense.”

Comment such as: ”Links to online resources would be helpful” indicate that learners also consider that additional online resources would be helpful if they were linked to LO.

Conclusions that can be made for the improvements of short cycle “Programming in Java” program from the given results are the following:

- Workshops must be organized more often and should be longer
- It is important to provide additional resources to LOs or assigned problems, such as Internet resources and
- The extent of the course and its intensity should be re-examined.

Finally, Table 11 shows the achieved results (in ECTS) of students fo SCHE Programming in Java, enrolled 2nd of October 2017 with duration of 12 months. The results are taken on 11th of March 2019.

		Acheived ECTS per course of short program "Programming in Java", as on 11.3.2019														
	Student	KI101	KI102	KI103	KI104	KI105	KI201	KI202	KI203	KI204	KI205	KI206	KI301	KI401	KI402	Total
1	Zorana Milojković	4	3	5	3	4	4	4	4	7	4	5	6	4	3	**60**
2	Vesna Strahinić	4	3	5	3	4	4	4	4	7	4	5	6	4	3	**60**
3	Dejan Milić	4	3	5	3	4	4	4	4	7	4	5	6	4	3	**60**
4	Katarina Ristanović	4	3	5	3	4	4	4	0	7	4	5	6	4	0	**53**
5	Bojana Domazet	4	3	5	3	4	4	4	0	7	0	5	6	4	0	**49**
6	Branislav Manojlović	4	3	5	3	4	0	0	0	7	4	5	6	4	0	**45**
7	Nataša Ljubisavljević	4	3	5	3	4	4	4	0	0	0	0	0	0	0	**27**
8	Đorđe Ristić	4	3	5	3	0	0	0	0	0	0	0	0	0	0	**15**
9	Goran Grozdanić	4	3	5	0	0	0	0	0	0	0	0	0	0	0	**12**
10	Branislav Tošić	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
11	Marina Šćepanović	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

As it can be seen, only 3 of 11 trainees got on time 60 ECTS. Two trainees left the program at the beginning.

5. CONCLUSION

This paper deals with evaluation of a short cycle program "Programming in Java" implemented as a pilot program of PT&SCHE project and aims to provide one critical source of information in the order for the future improvement of courses, curriculum, and instructor's pedagogic efforts. Today, everyone can find all possible information on the Internet. However, if the learning process is routed and controlled by an appropriate technology and methodology as implemented in the short program "Programming in Java" realized as pilot program of PT&SCHE project on BMU, then the learning process is much faster, easier and more effective as shown in the conducted evaluation. The achieved results might suggest that the curriculum was too demanding for trainees, but it will allow them to find a job of Junior Java Programmer more easily, as they have a needed set of knowledge and skills. BMU is aiming to satisfy employers and their expectations. Trainees must understand that programming is a serious and demanding job. They need to be ready for serious tests when future employers will ask them to demonstrate what they can perform.

Therefore, future analyzes and improvements will be directed at what is considered by trainees as a lack in the current implementation of program, such as the intensity of the course with the increasing of the work with instructors.

REFERENCE LITERATURE

Higher Education Act (2017), The Higher Educatio Act, Republic of Serbia, “, Službeni glasnik RS, No. 88/17, Belgrade, 29 September 2017.

EQF (2008), The European Qualifications Framework for Lifelong Learning (EQF)“, European Commission, Office for Official Publications of the European Communities, Luxembourg, 2008, Available: https://www.eurashe.eu/. [Accessed: 24-Sep-2018].

“European ICT Professional Profiles” (2017), CEN ICT Skills Workshop „European ICT Professional Profiles in action“, INTERIM REPORT, June 2017

e-CF(2014), “The European e-Competence Framework (e-CF) version 3.0 – A common European Framework for ICT Professionals in all industry sectors. CWA 16234:2014 Part 1. © CEN, http://www.ecompetences.eu/e-cf-3-0-download/ [Accessed 10/03/19]

Domazet (2017), Development of curriculum of the pilot online SCHE program PROGRAMMING IN JAVA, Deliverable 5.1.1, Partners Technical Report, Erasmus + project "Introduction of part-time and short-cycle studies in Serbia/PT&SCHE", 20.07.2017, www.pt-sche.metropolitan.ac.rs

The Foundation ICT BOK (2015), The Foundation ICT Body of Knowledge, Version 1, 22 February 2015, a report prepared for the European Commission, DG Internal Market, Industry, Entrepreneurship and SMEs by the Service Contract: e-Skills: Promotion of ICT Professionalism in Europe | No 290/PP/ENT/CIP/13/C/N01C011 prepared by Capgemini Consulting and Ernst & Young, https://ec.europa.eu/docsroom/documents/21342/attachments/1/translations/en/renditions/pdf

SWEBOK 3.0 (2014), SWEBOK 3.0 – Guide to the Software Engineering Body of Knowledge, Editors Pierre Bourque, École de technologie supérieure (ÉTS) and Richard E. (Dick) Fairley, Software and Systems Engineering Associates (S2EA), IEEE Computer Society, https://www.computer.org/education/bodies-of-knowledge/software-engineering

Computer Science Curricula 2013, Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, December 20, 2013, The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society, https://dl.acm.org/citation.cfm?id=2534860