LUT/LENS/Software Engineering

Compiled by Associate professor Jussi Kasurinen and Professor Ajantha Dahanayake

Summer Period 2019 Study Guide

For convenience, changes to the “normal course” implementation are **bolded.**

Document released 15.3.2019

Table of Contents

Notes on Summer period arrangements and study blocks 3
On registration 4
More information 5
CT60A2411 Olio-ohjelmointi (Object-Oriented Programming), 6 cr 6
CT60A4303 Tietokantojen perusteet (Basics of database systems), 3 cr 8
CT60A7650 Database Systems Management, 3 cr 10
CT10A0400 Tekniikan kandidaatin työharjoittelu (Work Internship in Bachelor's Degree), 2 cr 12
CT10A4000 Kandidaatintyö ja seminaari (Bachelor's Thesis and Seminar), 10 cr 14
CT30A3370 Käyttöjärjestelmät ja systeemiohjelmointi (Operation Systems and System Programming), 6 cr 16
CT70A9100 Self-Study: Software Development Skills, 3-12 cr 18
CT30A3202 WWW-sovellukset (Web Applications), 6 cr 20
CT60A0202 Ohjelmoinnin ja data-analytiikan perusteet (Introduction to Programming and Data-analytics), 6 cr 22
CT60A2500 C-ohjelmoinnin perusteet (Principles of C-programming), 3 cr 24
CT60A4160 Ohjelmistotestauksen periaatteet, 3 cr 25
CT60A5400 Fundamentals of Game Development, 6 cr 26
CT70A3000 Software Maintenance, 6 cr 28
CT10A9520 Research Project in Software Engineering, 1-10 cr 30
CT10A7004 Sustainability and IT, 6 cr 32
CT70A9000 Geographic Information Systems, 6 cr 34
CT70A9200 Workshop for Scientific Writing, 2 op 36

Notes on Summer period arrangements and study blocks

During the summer period 2019, for the students of the degree programs in software engineering LUT Tietotekniikka is offering the following course packages, called blocks. The general concept is that the programs offer at least 15 summer study credits regardless of the stage of the studies to all existing students, towards either on B.Sc. or M.Sc. level degree. The first block is for the first year students only, whereas blocks 2 and 3 for the Bachelor's students mainly on the second or third year. Block 4 is for the Master's level students, and those Bachelor's level students, who cannot take anything else from Blocks 1-3 but need more credits for the summer time study benefits:

Block 1: Summer camp courses for the year BSc1 students

Courses	@	Code	ECTs
Olio-ohjelmointi	Summer camp, LUT	CT60A2411	6
Tietokantojen perusteet	Summer camp, LUT	CT60A4303	3
Database Systems Management	Summer camp, LUT	CT60A7650	3
Tekniikan kandidaatin työharjoittelu		CT10A0510	2

To take this course package student is required to enroll and apply to the separate "summer camp" position at the LUT university. Students enrolling to this Block are expected to take all of the Block 1 courses during the Summer period, and are heavily recommended to not take further Summer period courses from Block 3 unless they are need for study benefits.

Block 2: Thesis module for 2nd or 3rd year BSc students

Course	@	Code	ECTs
Kandidaatintyö ja seminaari	LUT	CT10A4000	10
Kandidaatintyö ja seminaari	Remote	CT10A4000	10
Workshop for Scientific Writing	LUT- One day event	CT70A9200	2

Add courses from Block 3 or 4 to get summer period student benefits if needed.

The Bachelor's thesis course can be completed as locally arranged course, or as long-distance studies. Please enroll to the course, and contact the lecturers if you cannot participate to the teaching events at LUT Lappeenranta campus.

Block 3 - BSc level courses offered to all Bachelor's level students at LUT Software Engineering during the Summer period.

Course	Code	ECTs
Käyttöjärjestelmät ja systeemiohjelmointi	CT30A3370	6
Olio-ohjelmointi	CT60A2411	6
Self-Study: Software Development Skills	CT70A9100	3-12
WWW-sovellukset	CT30A3202	6
Ohjelmoinnin ja data-analytiikan perusteet	CT60A0203	6
C-ohjelmoinnin perusteet	CT602500	3
Ohjelmistotestauksen periaatteet	CT60A4160	3

Block 4: MSc level courses offered to all LUT Software Engineering students during the Summer period.

Course	Code	ECTs
Fundamentals of Game Development	CT60A5400	6
Research Project in Software Engineering	CT10A9520	3-10
Software Maintenance	CT70A3000	6
Sustainability and IT	CT0A7004	6
Geographic Information Systems	CT10A9000	6

On registration

There are three ways to register for the summer period courses.

1. **Registration requires applying to <u>Summer camp</u>** means that to take this course, you need to apply for a summer camp training position with the university. The summer camp requires mandatory presence as in an office workplace due to it trains simultaneously students to work life and co-operation. The summer camp program starts 20th May 2019 and ends 19th July 2019. Passed courses and filled presence and activity requirements gives accreditation from the internship.

   To register to summer camp, please send your transcript (taken from WebOodi), CV, and open form application letter for the "LUT Software engineering summer school trainee" position to summer camp coordinator Pirjo Kuru (pirjo.kuru@lut.fi) via email. Please, note that summer camp participants amount is limited to maximum of 25 students. Summer camp is available also for the students of the computational engineering.
   For further instructions contact Pirjo Kuru via email. The last day for sending your application letter and the appendices is **Friday 29th of March 2019 15:59**. All students who are accepted to the summer camp are contacted personally via email before the 14th of April.

2. **Registration via Weboodi.** To register to this course, register via Weboodi as you would do during the normal academic year. Weboodi registration for the summer period is open between **12.3. - 12.5.2019**.

3. **Contact Teacher in Charge.** This course requires the participant to register to the Weboodi, and then contact the Teacher in Charge for an individual assignment, or detailed instructions.

Also observe, that the summer courses have varying start and end dates (please see course descriptions attached), so observe the course Moodle pages for additional instructions and timetables!

This also means that on all courses, ensure that you can access the course Moodle page, and contact the course lecturer if unable to do so.

All Summer period courses are dated so that they are counted towards courses completed during the academic year of 2018-2019.

More information

Please contact Assoc. Prof. Jussi Kasurinen (jussi.kasurinen@lut.fi) for general inquiries,

Course administrator Pirjo Kuru (pirjo.kuru@lut.fi) for the summer camp,

Or the teacher giving the course for course-related details.

CT60A2411 Olio-ohjelmointi (Object-Oriented Programming), 6 cr

Note	**There are two variants of this course: The Summer camp variant for the summer camp trainees, and self-study version for others.**
Year	B.Sc. (Tech.) 2
Period	**Summer period, May 1 – July 31, 2019.** **Summer camp participants: registration requires applying to summer camp.** **Self-study participants: Registration via Weboodi, also Contact Teacher-in-Charge.**
Teaching Language	Finnish
Teacher(s) in Charge	Junior researcher, M.Sc. (Tech) Jiri Musto
Aims	At the end of the course students will be able to 1. Solve typical programming problems with object-oriented programming methods 2. Use Java and its features in programming 3. Read and describe Java code and UML diagrams 4. Utilize version control 5. Design basic graphical user interface.
Contents	Object-orientation, classes, inheritance, basics of modelling classes, principles of Java, basic data structures, abstract data types, exceptions, graphical user-interface.
Teaching Methods	**Summer camp:** Lectures 2 h, videos 8 h, **independent study, exercises, project work and participation to the summer camp, 142 h.** Training for the exam and exam 8 h. Total amount of work 160 h. Independent study supported by summer camp, contact hours and support events; announced via course Moodle pages.

	Self-study: Videos 10h, independent study, project work 142h. Training for exam and exam 8h.
Examination in Examination schedule	No
Examination in Moodle	Yes
Examination in Exam	No
Assessment scale and assessment methods	0 – 5. Exam 30%, exercises 25%, practical assignment 45%
Course Materials	Lecture slides and videos Eckel B (2006) Thinking in Java, 4th ed. Prentice Hall, Upper Saddle River, NJ, Thinking in Java, Herala A, Vanhala E, Nikula U (2015) Olio-ohjelmointi Javalla, versio 1.0. LUT Scientific and Expertise Publications/Oppimateriaalit-Lecture Notes Other material announced in the lectures.
Prerequisites	CT60A0220 C-ohjelmoinnin ja testauksen periaatteet, CT60A2500 and CT60A4160 or equivalent.
Places for exchange-students	No
Places for Open University Students	**Yes, max 15 in the self-study implementation**

CT60A4303 Tietokantojen perusteet (Basics of database systems), 3 cr

Note	**This course is for the Summer camp participants only. Non-summer camp registrations will be removed.**
Year	B.Sc. (Tech) 2
Period	**Summer; May 1 – July 31, 2019, registration requires applying to summer camp.**
Teaching Language	Finnish
Teacher(s) in Charge	DSc. Antti Knutas
Aims	At the end of the course the student will be able to: 1.Design and model relational databases 2.Understand how the evolution of relational algebra led to SQL databases 3.Model real world problems with ER and transform the ER model to relational databases 4.Understand and solve issues related to relational database design, such as optimization and normalization 5.Implement relational databases in practice and embed them in applications
Contents	Database systems. Database design. Object-centric modeling and ER-modeling. Specifying relation models. SQL and object languages. Perspectives into database design: How database is designed, how information is modeled, and what are information storage structures and access methods. Transforming ER models to relation model, and then to relation databases. The use of different file formats in different environments. Perspectives to database programming: queries and other operations, database management, e.g. triggers. Implementing databases in practice and how to use SQL databases from other programs.
Teaching Methods	Online lectures ja -exercises 13h, SQL-online course 20h **independent study, exercises, project work and participation to the summer camp, 33 h.** Preparing for exam 10h and online exam 2h. All 78h.

Examination in Examination schedule	No
Examination in Moodle	Yes
Examination in Exam	No
Assessment scale and assessment methods	0-5, project, weekly assignments, online exam. SQL syntax self-study, online (Viope)20% Weekly exercises20% Project assignment40% Online exam20%
Course Materials	Beynon-Davies, P.: Database Systems, Palgrave Macmillan, Third Edition, 2004. Foster, Elvis, C.: Database Systems A Pragmatic Approach, Apress, 2014. Lecture notes and other material assigned at the course.
Prerequisites	Basics of programming.
Places for exchange-students	No
Places for Open University Students	**No**

CT60A7650 Database Systems Management, 3 cr

Note	**This course is for the Summer camp participants only. Non-summer camp registrations will be removed.**
Year	B.Sc. (Tech.) 2
Period	**Summer; May 1 – July 31, 2019, registration requires applying to summer camp.**
Teaching Language	English
Teacher(s) in Charge	Post-doctoral researcher, D.Sc. (Tech.) Antti Knutas
Aims	At the end of the course students will be able to 1. Create a relational model and a relational database 2. Use relational algebra and relational calculus 3. Design a database application, data distribution, and architectures for data storage, retrieval, and administration of a database management system 4. Apply scalability, performance, security, and authorization 5. Demonstrate the knowledge of concepts and principles underlying the functioning of database management systems and maintenance.
Contents	Relational model and relational database design, Introduction to relational Algebra. Database applications, data distribution and architectures. Data storage and retrieval, data scalability, performance, security, authorization. Modeling and programing for semi-structured data, secondary storage management.
Teaching Methods	**independent study, exercises, project work and participation to the summer camp, 34 h.** Individual assignments, hands on team project assignment 44 h. Total 78 h. Independent study supported by summer camp, contact hours and support events; announced via course Moodle pages.
Examination in Examination schedule	No
Examination in Moodle	No

Examination in Exam	No
Assessment scale and assessment methods	0-5. Individual assignments = 50%. Project Assignment = 50%
Course Materials	• Ramez Elmasri, Shamkant B. Navathe (2015), Fundamentals of Database Systems, 7th Edition, Published by Pearson. ISBN-13: 978-0-13-397077-7 • A. Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom: Database Systems : The Complete Book, Pearson Prentice Hall 2nd Edition, 2009
Prerequisites	CT60A4303 Tietokantojen perusteet required
Places for exchange-students	**No**
Places for Open University Students	No

CT10A0400 Tekniikan kandidaatin työharjoittelu (Work Internship in Bachelor's Degree), 2 cr

Note	**The summer camp participants will receive a work diploma, which allows them to apply for these internship credits after the summer camp has ended. No separate registration, see UNI and Weboodi for details.**
Teaching Language	Finnish and English
Teacher(s) in Charge	D.Sc. (Tech.) Ossi Taipale
Aims	After the work environment internship, the student will be able to define and explain what paid work is, what is involved in working for an employer and what the basic rules of the world of work are from the employee's perspective, and further, how to act in a working community. After the work environment internship, the student reports the contents of the internship. The student obtains a certificate of employment and attaches it to the internship application.
Contents	The student obtains a (summer) job from the company, works as an employee, requests a certificate of employment and applies for the approval of the work as an internship for the Bachelor's degree. A full-time employment relationship of at least four weeks can be approved as an internship. The completion of the Bachelor's thesis is not accepted as an internship. An employment relationship that took place before the studies can be approved as an internship providing that it has not been accepted and included in any other previous degree.
Teaching Methods	Applying for a job and recruiting 10 h, tasks connected to starting an employment relationship (e.g. orientation, the rules of the employment relationship and the work place) 15 h, observing (while working) how the working community operates (e.g. how work/production is organized, supervision, the working manners of the working community/teams, the social environment of the work place) 22 h, a written internship report 5 h (2-3 pages). Total 52 h.
Examination in Examination schedule (Yes/No)	No
Examination in Moodle (Yes/No)	No

Examination in Exam (Yes/No)	No
Assessment scale and assessment methods	Pass/Fail, internship report 100%
Course Materials	From the workplace
Places for exchange-students? (Yes, number/No)	No
Places for Open University Students?(Yes, number/No)	No

CT10A4000 Kandidaatintyö ja seminaari (Bachelor's Thesis and Seminar), 10 cr

Year	B.Sc. (Tech.) 3
Period	**Summer period, May 1 – July 30 2019. Registration via Weboodi. Registration to this course requires the student to register to the CT70A9200 Workshop for Scientific Writing.**
Teaching Language	Finnish and English
Teacher(s) in Charge	**Associate professor Jussi Kasurinen, Associate professor Uolevi Nikula**
Aims	After the course students are able to prepare an independent work on some specific problems in the field of the degree program in Computer Science. Students plan, schedule and implements a project independently and reports and presents the results. Student demonstrates the ability to collect relevant references and scientific material with ability to identify what is usable information and what is not, simultaneously considering the context of the work and current situation in realms of research and practice in given topic. In the end of the thesis work, the student demonstrates the general value of their work and also are able to explain the usability of their work in practical context and its novelty in general. The intended Learning Outcomes for the course: 1. Demonstrated skill to produce thesis work following academic practices 2. Student can plan the work independently and produce a usable time table for the work 3. Student can critically limit the literature/source material search into meaningful context 4. Student produces academic work with justified methodology selection 5. Demonstrated use of data / information analysis and results/conclusions described based on the data 6. Student can define clear research questions
Contents	Independent planning and implementation project in the field of computer science / software engineering area, which follows good academic practices. The work and new knowledge is presented in course seminars, with a thesis process starting presentation and results presentation.
Teaching Methods	**Participation to the first lecture event (Announced via Moodle).** Main part of the course is independent work and seminars. Course includes participation into seminars, seminar presentations, initial report, Bachelor's thesis with maturity examination. Workload is mostly self-study 250 h and seminar work 10 h, which totals to 260 h.

Examination in Examination schedule	No
Examination in Moodle	No
Examination in Exam	No
Assessment scale and assessment methods	0 - 5. Bachelor's thesis 80-100 %, seminar presentations and proper maturity test (abstract of the work) included into the thesis 0-20%.
Course Materials	The relevant shared material shall be given and presented in the course starting lectures. Additional thesis specific material will be given in the guiding process of the thesis work. LUT's Bachelor's thesis Guidelines in Uni portal: https://uni.lut.fi/en/web/lut.fi-eng/instructions-and-regulations (UNI > Studies > Instructions and regulations.)
Prerequisites	Highly recommended, 100 ECTS credit points earned, before starting the thesis work.
Places for exchange-students	No
Places for Open University Students	No
Note	**Students completing this course locally at Lappeenranta during Summer 2019 will be assigned topics via priority list available in Moodle pages and will have weekly meetings. Students completing this course as long-distance participants please contact courses lecturers.**

CT30A3370 Käyttöjärjestelmät ja systeemiohjelmointi (Operation Systems and System Programming), 6 cr

Year	B.Sc. (Tech.) 2
Period	**Summer period, May 1 -July 30, 2019. Registration via Weboodi.**
Teaching Language	Finnish
Teacher(s) in Charge	Post-doctoral researcher, D.Sc. (Tech.) Jussi Kasurinen
Aims	After successful completion of the course, the student: 1. Understands how an operation system is built and how it works with application programs, 2. Can divide complex problems into smaller sub-problems, 3. Knows how a large programming project the size of an operation system is planned and executed, 4. Has acquired an overview of the structure of a computer system and the connections to algorithmics, computer architecture, operation systems, compiler programs and interpreters and to software production, 5. Understands the importance of systems programming in application development and in system maintenance, 6. Knows the basics of the operation system, 7. Can write Posix programmes using the C language 8. Can apply Posix libraries and system level functions in his/her programmes, 9. Can write Unix scripts.
Contents	Basic structures and functions of the operation system: file system, processes, memory management. Structure of the Unix system. C programming language and its programming environment and tools in the Unix system. Unix command decoder programming. Standard I/O-library, advanced I/O functions. System data and files. Processes, process management, interprocess relations. Braid ends and their management. Service processes. Interprocess communication. Signals and their management.
Teaching Methods	**Lectures 2h (time and place announced via Moodle)**, **online lectures 26h,** and preparation for lectures 28 h. **Self-study and assignments 80 h**. Preparation for exam 17 h. Exam 3 h. Total workload 156 h.
Examination in Examination schedule	No

Examination in Moodle	Yes
Examination in Exam	**No. Replaced by Moodle exam.**
Assessment scale and assessment methods	0-5, exam 50 %, exercises and assignment 50 %
Course Materials	N. Nisan & S. Schocken: The Elements of Computing Systems, MIT Press, 2005 W. Stallings: Operating Systems, 7th Edition, Pearson Education, 2012 A. Silberschatz, P. Galvin, G. Gagne: Operating System Concepts, Wiley, 2012 W. Richard Stevens and Stephen A. Rago: Advanced Programming in the UNIX Environment, 2nd edition, 2011. Ellie Quigley: Unix Shells by Example, 4th edition, 2010. William Stallings: Operating Systems: Internals and Design Principles, 7th Edition, 2011.
Prerequisites	CT60A0200 Introduction to Programming (Ohjelmoinnin perusteet), CT60A0210 Practical Programming (Käytännön ohjelmointi), BM40A0300 Data Structures and Algorithms (Tietorakenteet ja algoritmit).
Places for exchange-students	No
Places for Open University Students	This course has 1-5 places for open university students. More information on the web site for open university instructions.

CT70A9100 Self-Study: Software Development Skills, 3-12 cr

Year	BSc (Tech.) 2
Period	**Summer period, May 1 -July 30 2019. Registration via Weboodi.**
Teaching Language	English
Teacher(s) in Charge	Professor PhD Ajantha Dahanayake Asistant Professor DSc Antti Knutas
Aims	1. Develop practical skills for software development 2. Learn the best practices and approaches of software development 3. Develop the skilled expected in industry to work as a software developer.
Contents	This course aims give students a chance to create unique projects with a hands-on approach. The course guides students to find their interest in software engineering skills and to help each student find their desired path in software developing in the future. The course consists of four (4) modules: frontend, backend, mobile apps, full stack - from which a student chooses min. one (1) module to complete. One module completed is equivalent to 3 ECTS cr. <u>Front-end</u> The Goal is to make a responsive webpage using html, CSS and a little JavaScript. These are the basic tools to make today's web-frontend. Students may use Bootstrap or animations in addition. The project focuses only on the layout, styles and the overall structure of the page. <u>Mobile</u> The Goal is to make an Android app with Android Studio. The app should have basic functionality with buttons and views. This module aims to teach the basics of mobile development. <u>Backend</u> This module aims to teach the student what a backend is and what is it for. In this module you will learn how to implement a restful API from scratch using node.js, mongoDB and express. <u>Full-Stack</u> The module gives the student basic understanding of full-stack development. The Goal is to create a basic front- and back-end and bundle them together as a complete system. The focus is to understand the bigger picture and how to actually bundle different software components together to create a working program. You will learn how to use MEAN-stack as a fullstack tool bundle to create an app from scratch.

Teaching Methods	Per Module: Self-study: Reading 14 h, practice work 20h, Project assignment 44 h Student must do the following to pass the course: - Go through the given materials in the chosen module (Series of videos) and provide a reflection in learning diary. - Make a self-reflecting learning diary. (What you did learn and do, what problems did you face and how did you overcome them.) - Create a similar project to the one that you've been provided with in the learning materials. - Post everything in a Git repository and present the course work. - Give feedback on the course (mandatory)
Examination in Examination schedule	No
Examination in Moodle	No
Examination in Exam	No
Assessment scale and assessment methods	P/F, Learning Diary 40%, Final project 60%
Course Materials	Available online (Moodle)
Prerequisites	CT30A2803 User Interfaces and Usability CT60A0202 Ohjelmoinnin ja data-analytiikan perusteet
Places for exchange-students	max 5
Places for Open University Students	No

CT30A3202 WWW-sovellukset (Web Applications), 6 cr

ear	B.Sc. (Tech) 3
Period	**Summer period, May 1 -July 30 2019. Registration via Weboodi.**
Teaching Language	Finnish
Teacher(s) in Charge	D.Sc. Antti Knutas
Aims	At the end of the course the student will be able to: 1. Create web-based software products 2. Understand the evolution of web software and how it led to current online environment 3. Design and implement complex software systems using web-based software and APIs 4. Understand and solve issues related to web environment, such as caching and security 5. Solve real world problems and design online web systems using requirements based on these problems
Contents	WWW –application architectures and standards. Programming languages and APIs for creating interactive server and client software (e.g. JavaScript, PHP, AJAX). Efficient management of web-based software and publication. The course is programming intensive.
Teaching Methods	Online lectures and exercises 16 h, **independent self-study and assignments, 140h.** Total 156 h.
Examination in Examination schedule	No
Examination in Moodle	No
Examination in Exam	No
Assessment scale and assessment methods	0-5, project, weekly assignments, independent assignments. Written assignments 10% Weekly exercises 30% Practicum 50%

	Independent self-study material 10%
Course Materials	Crockford, D. (2008). JavaScript: The good parts. Sebastopol (CA): O'Reilly : Yahoo! Press. Bramer, M. (2015). Web Programming with PHP and MySQL: A Practical Guide (1st ed. 2015.). Cham: Springer International Publishing. Babin, L. (2007). Beginning Ajax with PHP: From Novice to Professional. Berkeley, CA: Apress, Inc. Other material presented at lectures.
Prerequisites	Basics of programming and data-analytics (former Basics of programming). Basics of database systems.
Places for exchange-students	No
Places for Open University Students	max 5

CT60A0202 Ohjelmoinnin ja data-analytiikan perusteet (Introduction to Programming and Data-analytics), 6 cr

Note	This course is given only in Finnish and thus it is not suitable for students who do not understand Finnish properly.
Year	B.Sc. (Tech.) 1
Period	**Summer; May 1 – July 30 2019, registration via Weboodi.**
Teaching Language	Finnish
Teacher(s) in Charge	tutkijaopettaja, TkT Uolevi Nikula
Aims	At the end of the course students will be able to 1. Create small programs with the Python programming language utilizing all basic commands and structures like list and class 2. Structure the program in multiple functions and libraries to make the programs understandable, maintainable, and extendable 3. Develop Python programs that can read data distributed as CSV files, select data of interest in the files, and analyze basic characteristics of the data 4. Do basic testing to a program to assess its quality.
Contents	History of programming and the situation today; Programming in Python; Good programming style and program performance; Basics of data analytics from the programming point of view.
Teaching Methods	**Lecture recordings, 16 h (Schedule announced via Moodle). Self study 42 h, compulsory assignments and project 90 h,** Preparation for the exam 7 h and exam 3 h. Total of 156 h.
Examination in Examination schedule	No
Examination in Moodle	No
Examination in Exam	Yes , **announced via Moodle**
Assessment scale and assessment methods	0 - 5. Exam 30 %, project 30%, weekly assignments 40%.

Course Materials	The LUT Python programming manual, lecture material, other material announced on lectures.
Places for exchange-students	No
Places for Open University Students	max 30

CT60A2500 C-ohjelmoinnin perusteet (Principles of C-programming), 3 cr

Note	**This course is given only in Finnish and thus it is not suitable for students who do not understand Finnish properly.**
Year	B.Sc. (Tech.) 1
Period	**Summer; May 1 – July 30 2019, registration via Weboodi.**
Teaching Language	Finnish
Teacher(s) in Charge	Associate Professor, D.Sc. (Tech.) Uolevi Nikula
Aims	At the end of the course students will be able to 1. Create small programs with the C-programming language utilizing all basic commands, data structures, and libraries. 2. Structure the program in multiple functions and files to make the programs understandable, maintainable, and extendable. 3. Utilize pointers and dynamic memory allocation to create and manage linked lists. 4. Use make -program to manage program compilation. 5. Use version management system to manage files.
Contents	C-programming language, pointers and dynamic memory management, good programming practices, make and version management tools.
Teaching Methods	**Lecture recordings 7 h (Schedule announced via Moodle),** self study 14 h, compulsory assignments 48 h, Preparing for the exam 7 h and exam 2 h. Total of 78 h.
Examination in Examination schedule (Yes/No)	No
Examination in Moodle (Yes/No)	No
Examination in Exam (Yes/No)	Yes
Assessment scale and assessment methods	0-5. Exam 30%, project 30%, weekly assignments 40%.
Course Materials	C-kieli ja käytännön ohjelmointi osa 1, other materials announced at the lectures.
Prerequisites	Programming skills, e.g. CT60A0201 Introduction to programming.
Places for exchange-students? (Yes, number/No)	No
Places for Open University Students?(Yes, number/No)	max 30

CT60A4160 Ohjelmistotestauksen periaatteet, 3 cr

Note	**This course is given only in Finnish and thus it is not suitable for students who do not understand Finnish properly.**
Year	B.Sc. (Tech.) 1
Period	**Summer; May 1 – July 30 2019, registration via Weboodi.**
Teaching Language	Finnish
Teacher(s) in Charge	Associate Professor, D.Sc. (Tech.) Uolevi Nikula
Aims	At the end of the course students will be able to 1. explain the basic terms and concepts in software testing 2. do software testing in unit, integration, and system levels 3. use basic testing tools in testing and automate testing tasks 4. work in a testing team as a junior software tester.
Contents	Software testing techniques, levels, automation, tools, working as a tester in a software testing team.
Teaching Methods	**Lecture recordings 14 h (Schedule announced via Moodle)**, self study 14 h, compulsory assignments 41 h. Preparation for the exam 7 h and exam 2 h. Total of 78 h.
Examination in Examination schedule (Yes/No)	No
Examination in Moodle (Yes/No)	No
Examination in Exam (Yes/No)	Yes
Assessment scale and assessment methods	0-5. Exam 30%, project 30%, weekly assignments 40%.
Course Materials	Ohjelmistotestauksen käsikirja, Jussi Pekka Kasurinen, Docendo Oy, 2013. Purchasing the book is not necessary to complete the course, other materials announced at the lectures.
Prerequisites	Skills attained in courses like CT60A0201 Introduction to programming and CT60A2500 Principles of C-programming.
Places for exchange-students? (Yes, number/No)	No
Places for Open University Students?(Yes, number/No)	max 15

CT60A5400 Fundamentals of Game Development, 6 cr

Year	M.Sc. (Tech). 1
Period	**Summer period, May 1 – July 30. Registration via Weboodi.**
Teaching Language	English
Teacher(s) in Charge	Associate professor Jussi Kasurinen

Aims

Intended Course Learning Outcomes. At the end of this course students will be able to:

1. Conduct independent work in entertainment software engineering context.
2. Independently design and implement a small-scale game program with some industry-relevant platform.
3. Acquiring further knowledge concerning the taught game development tool.
4. Working as a productive member and as part of a team developing larger entertainment software product.

Contents

Applied software engineering course. The objective for this course is for students to learn how to use their software engineering knowledge in an entertainment software engineering context. With the selected game development tools, student is capable to independently design and develop a small game program on some modern game engine platform, or work as a part of a team developing a larger game product.

List of Topics: lectures and project works:

- Games as software products
- Basics of processes and models applied in the entertainment software industry
- Basics of the game development tools
- Introduction to game engines and their functions
- Basics of 3D objects
- Introduction to game development-related programming problem.
- Basics of artificial intelligence in entertainment software engineering context.
- Basics of sound engineering
- Gamification and Serious games.

Teaching Methods

Primary mode of work is assisted self-study. **Lectures 2 h (time and place announced via Moodle), Online recordings 6h,** Independent work and project assignments 148 h. Total 156 h.

Suitability for doctoral studies	Yes

Examination in Examination schedule	No
Examination in Moodle)	No
Examination in Exam	No
Assessment scale and assessment methods	0-5. Continuous evaluation (no exam) Project proposal and presentation 20% Individual project assignments (x2) 60% Peer review work on other project assignments 20%.
Course Materials	Based on the yearly implementation; the taught game engine tutorials and other materials given during the course.
Places for exchange-students	No
Places for Open University Students	15

CT70A3000 Software Maintenance, 6 cr

ear	M.Sc. (Tech) 1
Period	**Summer period; May 1 -July 30 2019. Registration via Weboodi.**
Teaching Language	English
Teacher(s) in Charge	Associate professor Jussi Kasurinen
Aims	1. Work as software developers in the context of an existing code base 2. Know the best practices of software maintenance, including modern technical automation, management of technical debt, coding standards, refactoring, and design patterns 3. Learn about software evolution 4. Know how to produce and use reusable software
Contents	In industrial practice, software developers are often confronted with already existing software systems that need to be maintained, reused or evolved. This requires specific skills to understand the design and implementation of an existing system and which parts need to be modified, to build software systems that are easier to maintain, and to design systems with reuse and evolution in mind from the very start. This course will thus study a variety of techniques, tools and methodologies to help building software systems that are easier to understand, maintain, reuse and evolve.
Teaching Methods	**Independent study, online videos and online exam 66h.** Reading assignments, project **and exercise** assignments 88 h. Total 156 h.
Examination in Exam	**No**
Examination in Moodle	**Yes.**
Assessment scale and assessment methods	0-5. Project, online exam. Individual weekly assignments 20% Teamwork project 20% Online exam 20% Final project 40%
Course Materials	April, A., & Abran, A. (2012). Software maintenance management: evaluation and continuous improvement. John Wiley & Sons. Other material listed in the course website.

Prerequisites	CT60A0202 Basics of programming (previosly CT60A0201) CT60A2411 Object-oriented programming CT60A5103 Software Engineering Models and Modeling (previously CT60A5102)
Places for exchange-students	Max 5
Places for Open University Students	Max 5

CT10A9520 Research Project in Software Engineering, 1-10 cr

Note	To do this course study the Moodle page for the course and follow the instructions given there. The details of the course like the topic, scope, and the time frame are agreed with the person responsible for the course.
Year	M.Sc. (Tech.) 1-2
Period	**Summer period; May 1 -July 30 2019. Registration via Weboodi. Student is also required to contact Teacher in Charge.**
Teaching Language	English
Teacher(s) in Charge	Professor, Ph.D. Ajantha Dahanayake
Aims	At the end of this course students will be able to: 1. Identify a research problem and a research gap in a selected domain 2. Demonstrate the ability to select proper research questions for the selected research problem 3. Identify and apply proper research methods for answering to the selected research questions 4. Identify and apply proper tools for implementing the research 5. Demonstrate the skills and competencies to carry our a research project 6. Demonstrate scientific skills to report the research work in academic manners
Contents	This course is either an individual or a pair based research project in a selected research area. Students discuss with their supervisors on the suitable research project and create a research plan prior to starting the project. Students plan, implement and report the project using scientific and academic approaches. Students can implement their projects separately or as a part of a research team in the laboratory.
Teaching Methods	Participation in the work of the research group, 1st-4th period **and summer**. Total 26-260 h.
Suitability for doctoral studies	Yes
Examination in Examination schedule	No
Examination in Moodle	No
Examination in Exam (Yes/No)	No

Assessment scale and assessment methods	Passed/failed. Research report and seminar presentation.
Course Materials	Literature related to the research topic, agreed with the supervisor of the work.
Prerequisites	CT10A9511 Research Methods in Software Engineering or comparable course.
Places for exchange-students	max 5
Places for Open University Students	max 5

CT10A7004 Sustainability and IT, 6 cr

Note	This course is meant only for the fulltime students of the software engineering programme.
Year	M. Sc. 1
Period	**Summer period, May 1 -July 30 2019, registration via Weboodi. Student is also required to contact Teacher in Charge.**
Teaching Language	English
Teacher(s) in Charge	Prof., D.Sc. (Tech.) Jari Porras
Aims	At the end of this course students will be able to: 1. Identify various sustainable development challenges in the surrounding society 2. Demonstrate the critical thinking and argumentation skills in the discussions of sustainable development challenges 3. Identify the possibilities of IT and especially software engineering in the sustainable development challenges 4. Apply IT and especially software engineering for sustainable development challenges
Contents	The course emphasizes the role and impact of IT field and especially software engineering in the sustainable development. The topic is covered through selected books and scientific articles. Students may be divided into small groups that will each study the topic.
Teaching Methods	This course follows flipped classroom approach. Introductory lectures are used for introducing the lecture material and dividing students into smaller groups. **Online assignments (announced via Moodle), independent study and independent work 156 h**.
Assessment scale and assessment methods	0-5 continuous evaluation (no exam) **Assignments 100%**
Course Materials	Murugesan S. & Gangadharan G.R.: Harnessing Green IT - Principles and practices, Wiley, 2012, 433 p Tomlinson B.: Greening through IT - Information Technology for Environmental Sustainability, MIT Press, 2010, 221 p

	A set of yearly changing scientific articles that will be announces at the moodle pages of the course.
Limitation for students	This course is meant only for the full time students of the software engineering programme
Places for exchange-students	max 10

CT70A9000 Geographic Information Systems, 6 cr

Note	Course is fully independent online learning, no mandatory lectures or attendance.
Year	M.Sc (Tech) 1-2
Period	**Summer period, May 1 - July 30, 2019. Registration via Weboodi.**
Teaching Language	English
Teacher(s) in Charge	Professor, Ph.D. Ajantha Dahanayake, M.Sc. (Tech), junior researcher Jiri Musto
Aims	At the end of this course students will be able to: 1. Describe the GIS terminology and concepts, and the special nature of spatial data and how they are different from non-spatial data. 2. Identify the key components of a GIS, including users, databases, software, and networks. 3. Use GIS software tools and publish maps 4. Create GIS using Oskari or Web technology 5. Conduct research in the field of GIS
Contents	Basics of geographical information systems. Conducting research related to geographical information, acquiring geographical data
Teaching Methods	Whole course is independent online learning: Moodle lessons 21h, quizzes 4h, Moodle exercises 7h, assignments 24h, projects 82h, extra studying 18h. Total 156h
Suitability for doctoral studies	Yes
Examination in Examination schedule	No
Examination in Moodle	No
Examination in Exam	No

Assessment scale and assessment methods	0-5 continuous evaluation (no exam) Quizzes 5%, assignments 25%, written project 30%, programming project 40%
Course Materials	• Campbell J, Shin M (2011) Essentials of Geographic Information Systems. Saylor Foundation • Lecture videos and slides • Other material mentioned in lectures
Prerequisites	JavaScript programming knowledge
Places for exchange-students	15
Places for Open University Students	15

CT70A9200 Workshop for Scientific Writing, 2 op

Year	BSc (Tech) 3
Period	**Summer period; May 1 – July 39, 2019. This course is mandatory for students taking CT10A4000 during the summer period.**
Teaching Language	**Finnish**
Teacher(s) in Charge	**Associate professor Jussi Kasurinen**
Aims	1. Able to formulate a research problem 2. Able to apply technical writing for a scientific report 3. Able to conceptualize and produce a scientific report 4. Able to formulate the references and citations for the scientific work
Contents	Students who are starting their BSc thesis work will participate in the workshop. Learn the process and formulation of the content, and the written presentation of a scientific report.
Teaching Methods	Hands on practice work, learning by doing. **Independent work and Skype consultation for long-distance participants.** 6h in class workshop, 12h written assignment, 22h reading, 12h preparation work.
Examination in Examination schedule	No
Examination in Moodle	No
Examination in Exam	No
Assessment scale and assessment methods	P/F, report writing, assignments, presentations 100 %
Course Materials	Will be made available during the workshop
Prerequisites	Completed at least 100 ECTs of BSc course work **Bachelor's thesis work started before start of this course (Spring 2019 or Summer 2019).**

Limitation for students	25 – Priority participation by SE Bachelor's degree students completing their Bachelor's thesis on Summer 2019, or have reserved topic from Spring 2019.
WebOodi enrollment for excercises, amount of groups	1
Places for exchange-students?	No
Places for Open University Students?	No