Faculty of Computer Science
University of Indonesia

CURRICULUM 2016 HANDBOOK

INTERNATIONAL UNDERGRADUATE PROGRAM IN COMPUTER SCIENCE

Faculty of Computer Science
University of Indonesia

CURRICULUM 2016

Curriculum Design

Curriculum 2016 applied to students enrolled in 2016 and after. The curriculum was designed for eight semesters (four years), divided into two periods:

- Period 1:
  Semester 1 – 5 at University of Indonesia (UI)
- Period 2:
  Semester 6 – 8 overseas at partner university

UI uses SKS (*Sistem Kredit Semester*) system to weight the course load. One SKS is equivalent to 1-hour lectures or 2-hour lab works followed by 1-2 hours scheduled activity and 1-2 hours independent activity per week. Normally, students take about 18-19 SKS per semester. One semester is equivalent to 14-16 weeks of academic activities.

Students must take 144 SKS (credit units) to earn undergraduate degree in Computer Science from UI. The 144 SKS is distributed as follows:

Semester 1	20
Semester 2	18
Semester 3	20
Semester 4	19
Semester 5	18
Semester 6-8	49
Total	**144**

In Semester 6-8, students must earn credits units (CU) to get bachelor degree from the partner university. Later the credits units they gain from the partner university will be transferred in order to obtain their bachelor degree in UI. Therefore, the total credit units should be sufficient to be transferred into 49 SKS (terms and conditions applied).

Curriculum Breakdown

	Code	Course Name	Sks	Ref. (page)
Semester 1				
1	UIST600141	Mathematics 1	3	11
2	UIST601111	Physics	3	-
3	CSGE601010	Discrete Mathematics 1	3	24
4	CSGE601020	Programming Foundations 1	4	42
5	CSCM6011150	Introduction to Digital Systems	4	35
6	UIGE600003	English	3	-
		TOTAL	**20**	
Semester 2				
1	CSCM602115	Mathematics 2	3	13
2	CSGE601011	Discrete Mathematics 2	3	26
3	CSGE601021	Programming Foundations 2	4	45
4	CSCM601252	Introduction to Computer Organization	3	33
5	CSGE602012	Linear Algebra	3	37
6	UIGE60001*	Religion	2	-
		TOTAL	**18**	
Semester 3				
1	CSGE602055	Operating Systems	4	41
2	CSGE602013	Statistics & Probability	3	50
3	CSGE602040	Data Structures & Algorithms	4	20
4	CSGE602022	Web Design & Programming	4	53
5	CSCM602241	Automata & Theory of Languages	4	9
6	UIGE6000**	Arts & Sports	1	-
		TOTAL	**20**	
Semester 4				
1	CSGE602070	Databases	4	22
2	CSCM602023	Advanced Programming	4	5
3	CSCM603130	Intelligent Systems	4	30
4	CSCM603154	Computer Networks	4	17
5	CSCM603125	Software Engineering	3	49
		TOTAL	**19**	
Semester 5				
1	CSCM603127	System Programming	3	52
2	CSGE603291	Scientific Writing & Research Methodology	3	46
3	CSCM603217	Numerical Analysis	3	39

	Code	Course Name	Sks	Ref.
4	CSCM603228	Software Engineering Project	6	28
5	CSGE614093	Computer & Society	3	15
		TOTAL	**18**	
Semester 6-8		*--overseas--*	**49**	

During the semester 6 - 8, UI still has three compulsory courses which the students need to take:

1. CSCM604142 Algorithm Design & Analysis 4 SKS (page 7)
2. CSCM603234 Data Science & Analytics 3 SKS (page 19)
3. CSGE604098 Internships 3 SKS (page 32)

Therefore they may choose to take similar courses in the partner university or they may go back to UI after they complete their overseas studies to take above courses.

Academic regulations concerning registration, academic transcript, evaluation of study, academic leave, grade standard, maximum credit allowed in each semester, academic honesty, etc are all explained in *"HIMPUNAN PERATURAN AKADEMIK 2008" handbook* issued by *Direktorat Pendidikan Universitas Indonesia*.

Faculty of Computer Science
University of Indonesia

Advanced Programming
CSCM602023

General Information

Course Title	**Advanced Programming**
Course Code	**CSCM602023**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Foundation of Programming2 , Web Design & Programming**

Course Description

The course is a further study of Foundations of Programming. It will provide techniques for programming in the large and some selected advanced and latest emerging topics. This course will prepare the students to latest technology for enterprise programming. This course not only provide lecturing and practical skill but more importantly, it will teach the students the experience of adapting their skill to the latest technology for programming in the large. This course will develops the students ability to design, implement, and maintain moderately complex, realistically-sized programs using an Agile software development methodology. It builds upon the basic programming techniques introduced in introductory programming unit and offers the first introduction to the implementation of more complex real-world programs.
Topics of the course include: Object Oriented Analysis & Design, Applying OO principle, unit testing, software packaging and deployment, scaling up software, design pattern, continuous integration, concurrency, client server, familiarity with cloud computing, web services.

Course Objectives

After successfully completing this course, students should be able to:
(1) Specify, design and test component of software as part of a bigger software
(2) Understand and able to use the features in a modern and popular programming language such as Java
(3) Conduct good practice in programming as team members such as project management and version control
(4) Write high quality and reliable part of software by providing well written, complete documentation and unit testing

References

(1) Summerfield, Mark. Python in practice: create better programs using concurrency, libraries, and patterns. Addison-Wesley, 2013.

Faculty of Computer Science
University of Indonesia

Algorithm Design & Analysis
CSCM604142

General Information

Course Title	**Algorithm Design & Analysis**
Course Code	**CSCM604142**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Data Structure & Algorithm**

Course Description

This course provides an understanding on how to design and analyse an algorithm to answer the programming problems focusing on two main issues i.e. Correctness and Complexity.
The topics are including Introduction to algorithms: bubble sort, insertion sort, selection sort, searching; Growth of functions; Algorithm analysis: worst-case, best-case, average-case; Divide and conquer; Quicksort; Mergesort; Recurrence relation: master method, method of substitution, recursion tree; Heap sort; Lower bound of comparison based sorting; Linear sorting: bucket sort, radix sort, counting sort; Order statistics: selection problem; Dynamic programming: LCS, Matrix-chain multiplication; Greedy algorithm: fractional knapsack, job scheduling, MST; Backtracking: 0/1 Knapsack; Backtracking, branch and bound; Graph algorithms: BFS, DFS, shortest path, maximum flow; Sorting networks, parallel algorithms; Approximation algorithms, NP-completeness; NP-completeness;

Course Objectives

At the end of the course, students are expected to:
(1) be able to design an algorithm to solve programming problems utilizing design strategies such as iterative, recursion, divide and conquer, dynamic programming, greedy approach, backtracking, branch and bound.
(2) be able to prove correctness of an iterative algorithm.
(3) be able to analyze complexity of an algorithm and able to represent the algorithm complexity using the standard notation for such complexity.
(4) be able to recognize the complexity limitation within a computational model and able to distinguish the various class of problems within that limitation.

References

(1) Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., Introduction to Algorithms (2nd edition), MIT Press, 2001.

Faculty of Computer Science
University of Indonesia

Automata & Theory of Languages
CSCM602241

General Information

Course Title	**Automata & Theory of Languages**
Course Code	**CSCM602241**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Discrete Mathematics 1, Discrete Mathematics 2**

Course Description

This course is designed to teach students the field of automata and theory of languages, which is a sequel of Discrete Mathematics. Through the course, students are expected to understand the underlying concept of theory of computation, the computer languages, and several abstract machines as models of computation as the language recognition.
Topics discussed in this course are finite state automata and regular language, pushdown automata and context-free language, Turing machine and recursively numerable language, decision problem and (un) decidability.

Course Objectives

After successfully completing this course, the students should be able to:
(1) understand the fundamental concepts of computation theory
(2) understand several abstract machines with their languages and expressions, and able to design and develop the machines
(3) understand the fundamental concepts of computation complexity

References

(1) John C. Martin. Introduction to Languages and the Theory of Computation, 3rd Ed. McGraw-Hill. 2003
(2) John E. Hopcroft, Rajeev motwani, Jeffrey D. Ullman. Introduction to Automata Theory, Languages and Computation, 2nd Ed. Addison-Wesley. 2003
(3) Michel Sipser. Introduction to the Theory of Computation. PWS Publising. 1997

Faculty of Computer Science
University of Indonesia

Mathematics 1
UIST600141

General Information

Course Title	**Mathematics 1**
Course Code	**UIST600141**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

Calculus is one of the supreme accomplishments of the human intellect. Calculus provides tools and methods for analyzing and solving problems of change and motion. It introduces two new operations called differentiation and integration which are fundamental. Consequently, Calculus today continues to serve as the principal quantitative language of sciences and technologies.
The topics for this course are real numbers, complex number, inequalities and absolute value, one variable functions, graphs, function and its operations, limits and continuity of the functions of one variable, derivative and its applications, exponential and logarithmic functions, the notion of the integration, the techniques of integral calculations, the applications of the definite integral, and techniques of integrations.

Course Objectives

Upon successful completion of his course, students should have the following abilities.
(1) Students should be able to solve problems related to applications of derivative: rate of change, characterize the behaviors of functions based on its derivative, and locate the extremum of a continuous function over a closed interval.
(2) Students should be able to show the relation between two fundamental concepts of Calculus: differential and integral.
(3) Students should be able to calculate the area of a plane between two simple curves, arc length, and volume of solids obtained by rotating a curve over a coordinate axes.

References

(1) D. Varberg & E.S Purcell, 9th ed, Calculus, 2007, Prentice-Hall
(2) G.B Thomas & R.L Finney, Calculus and Analytic Geometry, 9th ed, 1996, Addison-Wesley

Faculty of Computer Science
University of Indonesia

Mathematics 2
CSCM602115

General Information

Course Title	**Mathematics 2**
Course Code	**CSCM602115**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Mathematics 1**

Course Description

This course teaches the advance topics of Calculus as The Definite integral; Applications of Integral; Trancedental Functions; Techniques of Integration; Indeterminate form and nonproper integral; Infinite sequences and series; Infinite sequences and series; Vector and Geometry Space; Vector and Geometry Space; Vector and Geometry Space; Multiple integral; Multiple integral; Further Applications of Integral

Course Objectives

Upon successful completion of his course, students should have the following abilities.
(1) Students should be able to solve problems related to applications of derivative: rate of change, characterize the behaviors of functions based on its derivative, and locate the extremum of a continuous function over a closed interval.
(2) Students should be able to show the relation between two fundamental concepts of Calculus: differential and integral.
(3) Students should be able to calculate the area of a plane between two simple curves, arc length, and volume of solids obtained by rotating a curve over a coordinate axes.

References

(1) Varberg, Dale E., Edwin Joseph Purcell, and Steven E. Rigdon. Calculus with Differential Equations. 9th Edition. Pearson/Prentice Hall, 2007.

Faculty of Computer Science
University of Indonesia

Computer & Society
CSGE614093

General Information

Course Title	**Computer and Society**
Course Code	**CSGE614093**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**(minimum credits of 100 SKS)**

Course Description

This course aims to raise the student awareness and sensibility to various social and economic problems regarding the implementation of computer technologies in daily lives. The students will be given some issues related to IT and they need to analyze the issues and recommend some solutions from their point of view as a computer science student.

Course Objectives

The objectives of this course are as following:
(1) hold a discussion session to analyze and recommend some solutions in the social, ethical and professional issues
(2) do some general surveys to see the impacts of technology especially information technology and stimulate the reflects of social and ethical issues
(3) push the students to assess the issues based on the values on their profession
(4) develop professional responsibility

References

There is no suggested textbook. Suggested reading materials will enrich students' understanding:
(1) K.W. Bowyer, Ethics and computing, 1996
(2) JA Senn, Information technology in business, 1995
(3) MA Arbib, Computers and the cybernetic society, 1983
(4) L. Long, Computers and information system, 4th ed., 1994
(5) Current newspapers, magazines, and other newsletters

Faculty of Computer Science
University of Indonesia

Computer Networks
CSCM603154

General Information

Course Title	**Computer Networks**
Course Code	**CSCM603154**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Operating System**

Course Description

The course is an introduction to the principles of the Internet based computer networks. It emphasizes on the top down approach of the Internet model, starting from the application layer on the top. It works its way down toward the transport, network, data link, and physical layers.

Course Objectives

The main objectives of this course are to understand about the basic concepts of communication networks, protocols and their works.
After the completion of this course, the students will be able to
(1) estimate the delay of a network
(2) analyze the user traffics
(3) design a reliable transport protocol
(4) design and troubleshoot a network with an IP basis

It is expected that upon successful completion of the course, students will be able to:
(1) Compare and contrast the services provided by each layer of an Internet based computer networks, i.e. the application, transport, network, data link, and physical layers.
(2) Construct an Internet based computer networks from the five layers with a top-down approach.

References

(1) James F Kurose & Keith W Ross, Computer Networking A Top-Down Approach, 6th Edition, Pearson Addison-Wesley, 2013

Faculty of Computer Science
University of Indonesia

Data Science & Analytics
CSCM603234

General Information

Course Title	**Data Science & Analytics**
Course Code	**CSCM603234**
Course Load	**3 credit hours (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Statistics & Probability, Databases**

Course Description

This course provides basic principals, techniques, and tools that are used in data science and analytics in order to extract the information or knowledge from a data. The pricipals and techniques are discussed based on various fields such as statistics, probability, database, machine learning and other computer science fields.
Some important concepts are data collection and integrations, exploratory data analysis, statistical inferences, Bayesian modelling, and data visualization. Ethical aspects, privacy and security in data science and analytics will also be discussed during the course. This course emphasizes on how to perform the integration of the given principals and techniques to solve problems in data science and analytics.
The topics are including :
(1) High Performance Data Analysis, Parallel Databases
(2) Parallel Query Processing, in-database analytics
(3) MapReduce, Hadoop, relationship to databases, algorithms, extensions, languages
(4) Key-value stores and NoSQL; tradeoffs of SQL and NoSQL
(5) Quantitative data analysis, basic concepts, experiment design, pitfalls
(6) Visualization, data products, visual data analytics
(7) Provenance, privacy, ethics , governance

Course Objectives

After successfully completing this course, the students should:
(1) be able to apply the correct computational methods in providing the computer-based solution as required and evaluate the proposed solution
(2) have a skill in processing the big data efficiently and fast, starting from data collection, data processing, data analysis and data visualization using the appropriate and current technology.

References

(1) O'Neil, C. and Schutt, R. Doing Data Science: Straight Talk from the Frontline, 2013, O'Reilly Media.

Faculty of Computer Science
University of Indonesia

Data Structures & Algorithm
CSGE602040

General Information

Course Title	**Data Structures & Algorithm**
Course Code	**CSGE602040**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Programming Foundations 1, Programming Foundations 2**

Course Description

The course teaches basic techniques for data abstractions, access algorithms, and manipulation of the abstract structure, as well as ad introduction to complexity analysis of space and time allocation in implementing the algorithms. The topics covered are: Abstract Data Type Concept, Linear Data Model: arrays ad dynamic lists, stacks and queues, sets, hierarchical Data model, Binary tree, Heap, Binary Search Tree, AVL-Tree, Red-Black Tree, B-Tree, Graph Model, Hashing, Searching and Tracing Algorithm.

Course Objectives

This unit aims to introduce the various ingredients required to program efficient solutions for real-world problems. Students are introduced to the various ways of dealing with collections of data as data structures, and are trained how to reason with them in terms of their efficiency given a specific problem. They will also be exposed to important constructs and patterns in programming, and how to compare two different programs through algorithm analysis. Finally, in the process, students will be encouraged to exercise good software engineering (and specifically, object-oriented) practices such as data abstraction, encapsulation, information hiding, and programming to an interface.

References

(1) Weiss, Mark Allen, Data Structures & Problem Solving using Java (3rd ed.), Addison-Wesley

(2) Michael T. Goodrich, Roberto Tamassia & Michael H. Goldwasser, Data Structures & Algorithms in Python, Wiley, 2013.
(3) Bradley N. Miller & David L. Ranum. Problem Solving with Algorithms and Data Structures Using Python, Franklin, Beedle & Associates,

Faculty of Computer Science
University of Indonesia

Databases
CSGE602070

General Information

Course Title	**Databases**
Course Code	**CSGE602070**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Programming Foundations 2**

Course Description

This course discusses the basic concept of database management that includes the aspect of modeling and design, language and facility, implementation and the use of database. Topics included are: architecture an concept of database management system (DBMS), file structure and organization, indexing, data modeling using entity-relationship model, data modeling using relational model, formal query language, relational algebra, relational calculus, SQL and QBE, functional dependencies, normalization of relational database, algorithm and relational database design process, query processing and optimization, transaction, concurrency control, database recovery and client-server database.

Course Objectives

The objectives of the course are:

(1) Given a real-life problem that will be represented into database application, students can design database application correctly by evaluating all related requirements.
(2) Given database queries, both simple and complex, students can use SQL to complete that query correctly.
(3) Given a logical database schema, students can decide appropriate data types for each field and constraints for each table and implement Data Definition Language (DDL) and Data Manipulation Language (DML) in the one of the popular Data Base Management System (DBMS).

References

(1) Elmasri dan Navathe, Fundamental of Database Systems, 4th Edition, Addison-Wesley, 2004
(2) Connoly, Thomas & Begg, Carolyn: Database Systems 4th edition, Prentice Hall, 2005.

Faculty of Computer Science
University of Indonesia

Discrete Mathematics 1
CSGE601010

General Information

Course Title	**Discrete Mathematics 1**
Course Code	**CSGE601010**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

Topics covered are: Propositional logic, First-order predicate logic, Proofs, Set and Functions, Integers and Division, Mathematical induction
Sequences, Progressions, the Growth of functions, Algorithms, Complexity of Algorithms, Number theory, Methods of proof, the Pigeonhole principle, Permutations, Combinations.

Course Objectives

The aim of this course is to prepare students to be able to demonstrate knowledge and understanding of the discrete mathematics appropriate for computer applications. Topics include propositional logic, predicate logic, set theory, functions, number theory, methods of proof, mathematical induction, the pigeonhole principle, permutations and combinations.
This course will introduce the student to concepts of discrete mathematics needed for the study of computer science. The course also serves to introduce students to rigorous mathematical problem solving, proof, and applications.
The student completing this course will be able to:
(1) Translate English sentence to propositions and vice versa
(2) Derive rigorous logical proofs given a set of premises using the rules of Propositional Logic and Predicate Calculus.
(3) Perform the basic operations of sets (union, intersection, difference, and complement) and apply the fundamental theorems of set theory to mathematical reasoning, probability, counting principles, and applications.
(4) Perform the basic operations of functions (inverse, composition)
(5) Contrast various methods of proof of classic mathematical theorems
(6) Compare the process of mathematical induction and the behavior of other types of sequences.

(7) Understand basic theories of integers and divisions such as prime numbers, gcd, lcm, modular arithmetic, and applications of the number theory
(8) Apply the pigeonhole principle in solving mathematical problems
(9) Calculate numbers of possible outcomes of elementary combinatorial processes such as permutations and combinations.

References

(1) Kenneth H. Rosen, Discrete Mathematics and Its Applications 6th edition, McGraw-Hill, 2007.

Faculty of Computer Science
University of Indonesia

Discrete Mathematics 2
CSGE601011

General Information

Course Title	**Discrete Mathematics 2**
Course Code	**CSGE601011**
Course Load	**3 hours credit**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

Topics covered are: Counting, Advanced Counting, Relations, Graph, and Tree.

Course Objectives

The aim of this course is to prepare students to be able to demonstrate knowledge and understanding of the discrete mathematics appropriate for computer applications. Topics include recurrence relations, generating functions, relations, graphs and trees.
This course will introduce the student to concepts of discrete mathematics needed for the study of computer science. The course also serves to introduce students to rigorous mathematical problem solving, proof, and applications.
The student completing this course will be able to:

(1) Model a real life problem using recurrence relation
(2) Solve linear homogeneous / non homogeneous recurrence relations with constant coefficients
(3) Understand the concepts of generating functions and use it to solve recurrence relations
(4) Understand the concept of relations and its applications
(5) Understand the properties of relations, closures of relation, equivalence relations, partial orderings and their applications in real-world problem
(6) Understand the concepts of graphs and able to model problems using graphs
(7) Understand the issues in graphs isomorphism, connectivity, Euler and Hamilton paths, shortest path problem, planar graph and graph coloring.
(8) Understand the concepts of trees and their applications
(9) Understand tree traversal and able to construct (minimum) spanning trees from known graphs

References

(1) Kenneth H. Rosen, Discrete Mathematics and Its Applications 6th edition, McGraw-Hill, 2007.

Faculty of Computer Science
University of Indonesia

Software Engineering Projects
CSCM603228

General Information

Course Title	**Software Engineering Project**
Course Code	**CSCM603228**
Course Load	**6 hours credit (6 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Databases, Software Engineering**

Course Description

The course provides the students with the experience to be actively involved in one semester Software Engineering Projects. It is expected that the students are able to synthesize and implement the knowledge from the related courses to develop an Software Engineering Projects.
The students work in a team, which consists of 4 to 5 students. All members must be actively participated in every role in Software Engineering Projects, i.e. project management (PM), business modeling and requirements (BR), analysis and design (AD), implementation (I), and testing (T). Each team member must participate in every activity of a software development life cycle, i.e. communication, planning, modeling, construction, and deployment.
A 14-week Software Engineering Projects is assigned to a team or several teams, depending on the scope and complexity of the project. The project problem is real, with real users.
During the project development, each team has several meetings with the users, as well as the internal team meetings. All meetings are recorded in the Minutes Of Meetings (MoMs), which will be considered in the grading.
Every team also presents their weekly progress in front of the course lecturers and assistants. Any problem in the development process is discussed in the weekly meeting.
At the end of the semester, all teams demonstrate their projects in front of the users, course lecturers, and assistants.
There are also two sessions of guest lecturers from reputable IT practitioners. They share their knowledge and best-practices in the management and development of Software Engineering Projects.

Course Objectives

The aims to provide the student with experiences in:

(1) Managing an Software Engineering Project, with all the related resources.
(2) Working effectively in teams.
(3) Working through a complete software development life cycle.
(4) Developing web-based applications.
(5) Practice the written and oral communication skills needed as IT professionals.

After successfully completing this course, the students should be able to:
(1) Apply the fundamental concepts and theories in software engineering.
(2) Locate, summarize, and evaluate information from various sources, including the guest lecturers.
(3) Identify and demonstrate appropriate use of basic project management processes and tools.
(4) Undertake an information technology project according to user requirements and project deadlines.
(5) Identify the characteristics of and apply techniques used in preparing and presenting, professional standard communication.
(6) Demonstrate professional standard written and oral presentation skill through appropriate tools.
(7) Identify the characteristics of effective teamwork and evaluate personal and peer contributions to team processes.
(8) Demonstrate their ability to work effectively in an IT team, manage their own time, study, and personal development.
(9) Develop codes with versioning and team standards
(10) Develop web applications using the latest web technologies.

References

(1) Harry Percival. Test Driven Development with Python. O'Reilly Media, 2014.
(2) Robert C. Martin, Clean Code: A Handbook of Agile Software Craftmanship, Prentice Hall, 2009.
(3) Pressman, Roger S. Software Engineering: A Practitioner's Approach 6th ed. McGraw Hill, Singapore, 2005.

Faculty of Computer Science
University of Indonesia

Intelligent System
CSCM603130

General Information

Course Title	**Intelligent System**
Course Code	**CSCM603130**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Discrete Mathematics 1, Statistics & Probability, Data Structures & Algorithms**

Course Description

This course introduces the basic concepts of artificial intelligence as well as a variety of approaches, techniques and methods that are often used. Lecture participants will be equipped with theoretical understanding and practical skills related to the development of software agents i.e. software that is smart and autonomous to resolve a problem

Course Objectives

At the end of the course, the students are expected to able to:

(1) Understand the basic concepts of intelligent software agents and autonomous system
(2) Applying state space search framework for modeling a complex problem, then design and implement a search agent that resolve the problem
(3) Designing and implementing local search and heuristic techniques to overcome the problem of compexity of the state space search
(4) Using symbolic logic as a knowledge representation language to model aspects of the real world, as well as understand the balance between representation and reasoning
(5) Understand the principles of probability and decision-makin based on learning from data

References

(1) Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach 3rd Edition. Prentice Hall 2010

Faculty of Computer Science
University of Indonesia

Internships
CSGE604098

General Information

Course Title	**Internships**
Course Code	**CSGE604098**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**(minimum credits of 100 SKS)**

Course Description

Through real world experience, this course gives a chance for students to enrich their knowledge and ability especially in problem solving for the real organization based on information technology.

Course Objectives

After completing this course, the students are expected to
(1) be able to communicate and collaborate with other co-workers in a spesific job about IT.
(2) Be able to apply the knowledges they gathered during their college to solve the problems in the industry where they do the internship for

References

-

Faculty of Computer Science
University of Indonesia

Introduction to Computer Organization
CSCM601252

General Information

Course Title	**Introduction to Computer Organization**
Course Code	**CSCM601252**
Course Load	**3 hours credit**
Course Level	**Undergraduate**
Prerequisite	**Introduction to Digital System**

Course Description

This course aims to teach students the fundamentals of sequential computer organization consisting these components: input, output, memory, and processor (control and datapath). The understanding of these concepts will also be supported with some programming exercises using low-level language e.g assembly language

Course Objectives

After completing this course, students are expected to:
(1) Understand the basic concepts of computer organization
(2) Understand how to execute the machine language instruction
(3) Be able to use assembly language to access the components in computer systems (input, output, memory, and processor (control and datapath))

After completing this course, students are expected to:
(1) Understand the notion of basic organization of a computer, different level of abstraction (from programming language to logic gates) and the technology trends that affect the organization and performance of a computer.
(2) Program in C-language as an introduction before bridging to assembly-language
(3) Understand the notion of stored-program computers; that processor manipulates numbers in the memory and that numbers can be interpreted as instructions or data depending on the mode of execution of the processor.
(4) Program using IA32 assembly language (generated by a C compiler), including generation of control constructs, implementation of procedures, implementations of data structures.
(5) Understand the notion of I/O devices, storage, and I/O performance, I/O access, DMA, I/O bus, and I/O interfaces.
(6) Understand the notion of interrupt, interrupt handling, multiple devices, exceptions, and the use of interrupt in operating systems.

(7) Understand the notion of fundamental operation within the datapath (data fetching/storing from/to memory, register transfers, arithmetic and logic operation), instruction execution, hardwired control, and microprogrammed control.
(8) Understand the notion of pipeline as a way to improve instruction execution, data dependency, pipeline and branching, and influence of pipelining on instruction set design.
(9) Understand the notion of semiconductor RAM memories, ROM, Cache memories, performance issues, and virtual memories.
(10) Understand the notion of architectural support for high level languages and operating systems.

References

(1) Patterson, David A., and John L. Hennessy. Computer organization and design: the hardware/software interface. Vol. 4. Elsevier, 2010.
(2) AVRStudio 4, ATMega Instruction Set, ATMega Datasheet 8515

Faculty of Computer Science
University of Indonesia

Introduction to Digital System
CSCM6011150

General Information

Course Title	**Introduction to Digital System**
Course Code	**CSCM6011150**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

This course is an introductory in digital design. Students will learn basic concepts of computer science and computational models such as binary representation, Boolean algebra, finite-state-machine and instruction-set processors. They also learn basic components for design on different levels of abstractions such as transistors, gates, flip flops, adders, multipliers, registers, memories and processors.
This course is devoted to understanding design technique of digital systems using high-level programming language (High Level Language). The language used is VHDL (VHSIC Hardware Description Language). VHSIC stands for Very High Speed Integrated Circuit, which is the term applied to the IC development project in the Department of Defense (DOD) United States. Better understanding of VHDL's elements, behavioral modeling techniques (behavioral), dataflow modeling and structural modeling are included in the lecture material. The elements of programming, including configurations concepts, subprograms, overloading, packages and libraries, also examples of modeling digital circuits common in digital systems will also be discussed in the course. Laboratory facilities support is a set of FPGA board and development tools from Xilinx

Course Objectives

Through the course, students are expected to understand the underlying concept of computers, i.e. digital systems. After completing the course, the students are expected to :
(1) Understand how computer system works in the lowest level
(2) Understand how computer components were built and how they formed the computer structures
(3) Understand internal collaboration among the components in computer
(4) Understand the relation between software and hardware
(5) Understand the consequences of internal structures into computer's performances

References

(1) Mano, M. Morris, Charles R. Kime, and Tom Martin. Logic and computer design fundamentals. Vol. 5. Pearson Education, 2015.
(2) Tan, Aaron Tuck Choy. Digital Logic Design. McGraw-Hill, 2004.
(3) Harris, David, and Sarah Harris. Digital design and computer architecture. 2nd Edition. Elsevier, 2013.

Faculty of Computer Science
University of Indonesia

Linear Algebra
CSGE602012

General Information

Course Title	**Linear Algebra**
Course Code	**CSGE602012**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

This course prepares the students to be able to solve problems about matrix algebra and vector space concept. This course teaches the students about logical reasoning and mathematical abstraction. Therefore, the students need to actively participate in the whole processes.
The topics included in this course are: Linear equation system, matrices, determinant, vector space, inner product space, Eigen value and Eigen vector, and linear transformation

Course Objectives

Linear Algebra is an important subject in undergraduate mathematics, particularly for computer science students. This course will present the main concepts and terminology of Linear Algebra that play an essential role in mathematics and in many technical areas of modern society, such as computer science. The purpose of this course is to teach the skills needed to recognize and translate the problem into mathematical model. Then, students should be able to apply the basic ideas of algebra in most everyday problems and be prepared for subsequent courses in computer science.
Based on the objectives of mathematics teaching, i.e., problem solving and intellectual sports, the objectives of this course are the following.

Upon successful completion of this course students should be able to:
(1) identify linear transformations, determine the standard matrix of a given linear transformation, interpret the properties of linear transformation on a plane and space.
(2) given a square matrix, students are able to find the eigen values and their related eigen vectors, and use the result to diagonalize the matrix
(3) given a system of linear equations, students are able to select the right method to solve or to calculate the least square approximation.

References

(1) Anton, Howard; Elementary Linear Algebra; 10th Edition, Jhon Wiley & Sons. Inc; New York, NY, 2011
(2) Lay, David C.; Linear Algebra and Its Apllication; 2nd Edition, Addison-Wesley Publ. Co.; Reading, Mass, 2000
(3) Johnson, Lee W., R. Dean Riess, Jimmy T. Arnold; Introduction to Linear Algebra, Addison Wesley, New York, NY, 2002

Faculty of Computer Science
University of Indonesia

Numerical Analysis
CSCM603217

General Information

Course Title	**Numerical Analysis**
Course Code	**CSCM603217**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Linear Algebra, Mathematics 2**

Course Description

The lectures will provide students with the basic knowledge of numerical methods to solve scientific and engineering problems; and tutorials will give the students an opportunity to participate in the discussion and share their knowledge and experience in some problems to be solved in Matlab computing environment. The module will be taught using a combination of online lectures with practical exercises to work on at students' own time and onsite tutorial/practical sessions. The practical sessions will give students hands-on numerical analysis and problem solving using Matlab as the programming environment.
Topics covered include: introduction which briefly explain the role of numerical methods in natural phenomena and industry; floating point operations; approximation by Lagrange, Power and Newton polynomial; linear system solutions which cover Gauss Elimination, pivot strategy, triangular factorization; solution for non linear equation; differential and integral approximation; the solution of ordinary differential equations which include Taylor, Euler, Runge-Kutta and predictor-corrector methods.

Course Objectives

This module aims to teach students fundamentals of scientific computing and problem solving using Matlab as a tool. By understanding the fundamentals of the numerical methods used in scientific computing, students will be able to solve scientific and engineering related problems efficiently in Matlab computing environment.

The objectives of this course are:
(1) Given scientific problem, the students are able to formulate the problem and analyze suitable method for solving the problem.
(2) After identifying suitable method for solving the problem, the students are able to use Matlab to obtain the solution and provide its graphical visualitation when it is necessary

References

Scientific Computing - An introductory survey, 2nd Ed, McGraw-Hill, Michael T. Heath, 2002

Faculty of Computer Science
University of Indonesia

Operating System
CSGE602055

General Information

Course Title	**Operating System**
Course Code	**CSGE602055**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Introduction to Computer Organization**

Course Description

This course discusses the organization, structure and concepts of computer operating system. Topics included in this course are: introduction (history, basic concept and structure), Process (model, inter-process, communication, scheduling), Memory Management (swapping, virtual memory, page replacement, paging, working set model, segmentation), File System (file and directory, implementation, security, protection), Input/Output, Deadlock, and Distributed System.

Course Objectives

This module aims to introduce students to the fundamentals of an operating system both designing as well as implementation issues. It will cover all fundamental principles such as processes management, interprocess communication, memory management, I/O management, file system management, implementation examples (GNU/Linux and MS Windows), as well as some additional topics.
After successfully completing this course, students should be able:
(1) To understand that an operating system is an important part of a computer system.
(2) To understand that the operating system module is the part of the computer science curriculum core.
(3) To understand and to be sensitive to the basic concepts of a modern operating system, such as modern computer structures, process management, memory, file systems and I/O, with no need of formal theoretical proofing.
(4) To understand several operating system concepts by the usage of Java Programming Language.
(5) To get experience in learning a specific operating system topic, as well as explaining the topic in a essay.
(6) Ability to relate the operating system concepts to real operating systems.

References

(1) A. Silberschatz, Operating systems concepts with Java 7th edition.
(2) A.S. Tannenbaum, Operating Systems Design and Implementation 3rd Edition, Prentice hall software series.
(3) Pengantar Sistem Operasi Komputer (monkey book - RMS & MDGR).
(4) William Stallings, Operating Systems, Prentice Hall 4th or later edition

Faculty of Computer Science
University of Indonesia

Programming Foundations 1
CSGE601020

General Information

Course Title	**Programming Foundations 1**
Course Code	**CSGE601020**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

This module aims to teach the fundamental concepts and techniques of computer programming through the object-oriented paradigm with the Python programming language. This module is taught using a combination of lectures and hands-on programming exercises. Each student sits at a desk equipped with a personal computer connected to the Internet.
Remember that programming cannot be taught in a linear way, but previous topics should be revisited at a deeper level subsequently.

(1) Describe the history of the programming language and its evolution from machine language to high-level language.
(2) Comprehend the basics of programming concepts and construct.
(3) Define the terms Class and Object, and understand the basic paradigm of object orientation.
(4) Have knowledge on the theory of control statements, such as selection and iteration.
(5) Properly organize large programs by creating methods and using those available in class libraries.
(6) Process data within a list or a table.
(7) Properly design reusable classes and realize the fundamentals of inheritance.
(8) Understand the relationship among classes within a hierarchy and reveal polymorphism within objects.
(9) Create error-free applications by properly handling exceptions.
(10) Accurately pass data to secondary storage devices, and understand the differences between text and binary data files.
(11) Design and build graphical user interface (GUI).
(12) Understand and construct program documentation.

Course Objectives

This course teaches the foundations of object-oriented application development. After completing the course, the students are expected to have a competency to apply the concepts of mathematics, science, and the fundamental of computer science to perform a sistematic, logical and computasional problem solving. They are also need to be able to explain the fundamental concepts of programming and apply that in problem abstraction so that the problem can be solved in a logical and computational way.

References

(1) Ljubomir Perkovic. Introduction to Computing using Python. 2nd Edition. John Wiley & Sons, 2015.
(2) Charles Dierbach. Introduction to Computer Science Using Python: A Computational Problem-Solving Focus. 7th Edition. John Wiley & Sons, 2012.

Faculty of Computer Science
University of Indonesia

Programming Foundations 2
CSGE601021

General Information

Course Title	**Programming Foundations 2**
Course Code	**CSGE601021**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Programming Foundations 1**

Course Description

This course is the second part of two-course sequence. The first course is Foundations of Programming 1. This course enhances the knowledge and experience of the previous course to build the required programming skill. It introduces new paradigms in standard modern programming language (such as lambda and generic programming where applicable to the chosen programming language). Topics may include: (1) Problem Solving and introduction to algorithm (2) Testing First (unit test) (3) More on methods, parameters, and lambda (anonymous function) (4) More on Recursion (tail recursion, recursion on simple data structure) (5) Basic sort & search algorithm (including approximation, newton-raphson) (6) Debugging (7) Introduction to profiling (8) Class Diagram (9) More on OOP (10) IDE support, version control (11) Modules, Packages (12) API (Application Programming Interface) (13) GUI (14) Introduction to Software Architecture

Course Objectives

This course teaches the foundations of object-oriented application development. After completing the course, the students are expected to have a competency to apply the concepts of mathematics, science, and the fundamental of computer science to perform a sistematic, logical and computasional problem solving. They are also need to be able to explain the fundamental concepts of programming and apply that in problem abstraction so that the problem can be solved in a logical and computational way.

References

Main References:

(1) Cay S. Horstmann. Big Java. 4th Edition. John Wiley & Sons, 2010.
(2) Paul Deitel, Harvey Deitel. Java How to Program. 8thEdition. Pearson. 2010

Additional References:

(1) Bruce Eckel. Thinking in Java. 4thEdition. MindView.

Faculty of Computer Science
University of Indonesia

Scientific Writing & Research Methodology CSGE603291

General Information

Course Title	**Scientific Writing & Research Methodology**
Course Code	**CSGE603291**
Course Load	**3 hours credits (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

This course is intended to enrich students' comprehension of the structure and execution of the written academic papers. Scientific Writing (SW) involves understanding the process of writing, techniques used in writing, and the writing itself. The development of writing should be an integrated approach of human-data-information-knowledge-tool interaction which may result in a sound and readable academic writing. It is cannot be considered as a one-sided or single component. It is an inter-related process, as it evolves to respond to changes in writers' mind.
The course will to a large extent deal with of how to develop writing so that its arrangement and understanding can be improved. Beside that, this course will explore the role of writing in the community-particularly in the scientific environment, that have moved from having a dominant verbal communication to having a core function in producing quality of information.

Course Objectives

After completing this course, the students should be able:
(1) to understand the basic process of academic writing
(2) to explore the various approaches in developing academic writing
(3) to exercise the development of scientific writing in the real world.

References

(1) Sekaran, Uma. "Research Methods for Business: A Skill-Building Approach". 2005
(2) Wilson Jr., E.B. "An Introduction to Scientific Research Methods"
(3) Christensen, Larry B. Experimental methodology, Pearson, 9th Edition, 2004
(4) Tan, Willie. Practical research methods. Singapore: Prentice Hall. 2002

(5) Myers, Michel D. Qualitative research in information systems: a reader. Sage pub, 2002
(6) Additional readings will be assigned during class session

Faculty of Computer Science
University of Indonesia

Software Engineering
CSCM603125

General Information

Course Title	**Software Engineering**
Course Code	**CSCM603125**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Programming Foundation 2**

Course Description

This course covers the software development cycle, which consists of planning, analysis, design, coding, testing and maintenance. In this course, students form a group and develop a project that will be conducted for one semester. Topics covered in this course are: software development methodologies, computer-aided software engineering (CASE tool); Software development project planning, problem analysis and user needs; Organizing software specifications; Basic principles of software design; Problems in coding, software quality assurance; Software quality measurement; Software testing, software maintenance

Course Objectives

The followings are the aims of this course:
(1) Students are able to choose and decide the proper methodology/process model for engineering a software based on specific condition.
(2) Students are able to model the software specifications at these stages: requirement gathering, analysis, and design.
(3) Students understand various software testing techniques.

References

(1) Pressman, Roger S., Software Engineering: A Practitioner's Approach, 7th Edition, Mc. Graw Hill International, USA, 2010.
(2) Sommerville, Ian, Software Engineering, 8th Edition, Pearson-Addison Wesley, England, 2007.
(3) Bentley, Lonnie D., Jeffrey L. Whitten, and Gary Randolph. Systems Analysis and Design for the Global Enterprise. 7th ed. Boston: McGraw-Hill Irwin, 2007.

(4) Dennis, Alan, et. al., System Analysis and Design with UML 3rd Edition, John Wiley & Sons, 2010.
(5) Larman, Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition, Pearson Education International, USA, 2005.
(6) Pfleeger, Shari Lawrence., and Joanne M. Atlee. Software Engineering: Theory and Practice. 4th ed. Upper Saddle River [N.J.: Prentice Hall, 2010.

Faculty of Computer Science
University of Indonesia

Statistics & Probability
CSGE602013

General Information

Course Title	**Statistics & Probability**
Course Code	**CSGE602013**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**None**

Course Description

This course gives the basics of statistic which help the students to conclude results of data interpretation and make appropriate decision. Topics included in this course are: introduction to elements for statistic, data interpretation, probability concept, random experiment, random variable, independent events, conditioned probability, Bayesian theorem, discrete and continuous distribution, sampling distribution, statistical hypothesis test, analysis of variance, and simple regression.

Course Objectives

After successfully completing this course, the students should be able to understand the concept of probability, random variable, descriptive statistics, and inferential statistics. The students should also be able to use these above concepts to solve problems about stochastic or combinatorics statistical phenomenon

References

(1) Introduction to Probability and Statistics for Engineers & Scientists, 3rd ed., Sheldon M. Ross, Elsevier, 2004
(2) A Modern Introduction to Probability and Statistics, Understanding Why and How, Frederik Michel Dekking et al., Springer, 2005
(3) Probability and Statistics for Computer Science, James L. Johnson, New Jersey: A John Wiley & Sons', 2008
(4) Probability, Statistics and Queueing Theory with Computer Science Applications, Arnold O. Allen, Academic Press, 1997

Faculty of Computer Science
University of Indonesia

System Programming
CSCM603127

General Information

Course Title	**System Programming**
Course Code	**CSCM603127**
Course Load	**3 hours credit (3 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Data Structure & Algorithm, Operating Systems**

Course Description

This course will deepen the concepts of operating system especially about GNU/Linux, with an emphasize on the practical ability to implement and modify the program under applications layer.
The students will be introduced to the different layers of implementation on operating system as process management, filing system, IPC, socket network, and kernel structures. Student exercises will be given on C/C++ language and some scripting tools to modify the files.

Course Objectives

After successfully completing this course, the students should be able to:
(1) be familiar with Linux file system, system processing and the structures of Linux kernel
(2) explain the different layers of operating systems and their implementations
(3) develop the utilities that is similar to the UNIX standard (e.g : mv, rm) using UNIX basic system call and its facilities to manipulate the screen (for text-based editors, menu-driven systems, forms, and etc)

References

(1) Beginning Linux Programming 4th Edition, Neil Matthew, Richard Stones, Wiley Publishing
(2) Kerrisk, Michael. The Linux programming interface. No Starch Press, 2010.
(3) Richardson, Matt, and Shawn Wallace. Getting started with raspberry PI. O'Reilly Media, Inc., 2012.

(4) Linux Manuals
(5) Signals Introduction
(6) GNU Coding Standards
(7) Kernel Compilations
(8) Rubini, Alessandro, and Jonathan Corbet. Linux Device Drivers. 2nd edition. O'relly, 2001.
(9) The Linux Kernel Module Programming Guide, Chapter 2 & 5
(10) Raspberry Pi Kernel Compilation (http://elinux.org/Raspberry_Pi_Kernel_Compilation)

Faculty of Computer Science
University of Indonesia

Web Design & Programming
CSGE602022

General Information

Course Title	**Web Design & Programming**
Course Code	**CSGE602022**
Course Load	**4 hours credit (4 SKS)**
Course Level	**Undergraduate**
Prerequisite	**Programming Foundations 1, Programming Foundations 2**

Course Description

HTML and CSS (Cascading Style Sheet) as the basic components in a web page will become the first topics taught in this course. For the next topic, this course will also discuss about the development of web appearance dynamically using Javascript. Javascript is also a fundamental for Ajax programming that will also be introduced in the end of this course.
This course will also explain about server programming using PHP and ASP language. To enrich to web presentation, this topic involves the development of image for a web page and dynamic presentation.

Course Objectives

It is expected that upon successful completion of the course, students will:
(1) be able to design and implement a website
(2) be able to develop a dynamic website
(3) understand how to create a web page that is easy to use using a programming paradigms in a client and server side

References

(1) Internet & World Wide Web How to Program, 3rd Editon, Deitel Deitel Goldberg