Computer Science & Information Management Data Science and Artificial Intelligence

Credits Requirements for CS&IM

Master program Requirements: 48 Credits		
	Thesis Option	**Research/Internship Option**
Required course	12 credits	12 credits
Elective course	14 credits	24 credits
Required Credits Options	22 credits	12 credits
TOTAL	**48 credits**	**48 credits**

Credits Requirements for DS&AI

Master program Requirements: 48 Credits		
	Thesis Option	**Research/Internship Option**
Required course	14 Credits	14 Credits
Elective course	11 Credits	21 Credits
Seminar (Pass/Fail)	1 credit	1 credit
Required Credits Options	22 credits	12 credits
TOTAL	**48 credits**	**48 credits**

Credits Requirements for Minor/Exchange

Master program Requirements: 48 Credits		
	Thesis	**Research/Internship**
Course work	24 credits	24 credits
Minor/Exchange		12 credits
Required Credits Options	22 credits	12 credits
TOTAL	**48 credits**	**48 credits**

Credits Requirements for Doctoral

Doctoral Program Requirements : 84 Credits	
Coursework	12 credits
Dissertation	72 credits
TOTAL	**84 credits**

Required Courses

Computer Science (CS)

- AT70.02 Data Structure and Algorithms
- AT70.03 Theory of Computation
- AT70.07 Programming Languages and Compilers
- AT70.12 Web Application Engineering

Area of specialization in Software Engineering

- AT70.18 Software Architecture Design
- AT70.19 Software Development and Quality Improvement
- AT71.05 Information Systems Development and Management

Information Management (IM)

- AT71.05 Information System and Development Management5
- AT82.02 Data Modeling and Management
- AT82.04 Business Intelligence and Analytics
- AT82.09 Human Computer Interaction and Information Visualization

Data Science & Artificial Intelligence (DS&AI)

- AT82.01 Computer Programming for Data Science and Artificial Intelligence
- AT82.02 Data Modeling and Management
- AT82.03 Machine Learning
- AT82.04 Business Intelligence and Analytics
- AT82.05 Artificial Intelligence: Natural Language Understanding
- AT83.13 Internship 8 weeks

Master program - Minor DS&AI

- AT82.02 Data Modeling and Management
- AT82.03 Machine Learning
- Elective 2 Courses in DS&AI

AT70.02 Data Structures and Algorithms, 3(3-0) Semester: August

Course Objective: The objective of this course is to provide the process by which a computer solves a problem. Moreover, an algorithm is tightly coupled with how information manipulated by it is organized, in particular, its data structures. Using a computer to tackle a problem, therefore, entails designing an efficient algorithm as well as appropriate data structures. In this course students will learn the methodology of designing algorithms and data structures given a problem to solve. They will learn as well to analyze and compare algorithms in terms of time and space used.

Learning Outcomes :
The students on the completion of this course would be able to:

- Given a problem, be able to understand the issues to consider in solving it on a computer.
- If the problem is solvable, be able to design an algorithm and its data structures for this purpose.
- Be able to apply a repertoire of standard algorithm-design techniques as well as data structures.
- Understand the particular design requirements of different problem domains.
- Given an algorithm, be able to analyze its time and space complexity.

Prerequisite: None

Course Outline:

I. Foundations
  1. Asymptotic Analysis
  2. Recursion and Recurrences

II. Randomized Algorithms
  1. Indicator Variables
  2. Probabilistic Analysis

III. Sorting
  1. Quicksort
  2. Sorting in Linear Time
  3. Order Statistics

IV. Hashing
  1. Hash Tables
  2. Hash Functions
  3. Universal Hashing
  4. Perfect Hashing

V. Balanced Search Trees
  1. Red-black Trees
  2. B-Trees

VI. Advanced Design Techniques
  1. Dynamic Programming
  2. Greedy Algorithms

VII. Graph Algorithms
  1. Shortest Paths
  2. Maximum Flow

VIII. Polynomials and the FFT
  1. DFT and FFT
  2. Efficient FFT Implementation

IX. String Matching
  1. Rabin-Karp Algorithm
  2. String Matching with Finite Automata
  3. Knuth-Morris-Pratt Algorithm

X. Other Topics
  1. RSA
  2. Edit Distance
  3. NP-Completeness

4. Linear Programming

Laboratory Session(s): None

Learning Resources: Textbook.

Reference Textbook(s):

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. (2009). **Introduction to Algorithms**, Third Edition (3rd ed.). The MIT Press.

Robert Lafore. (2002). **Data Structures and Algorithms in Java,** Second Edition (2nd ed.). Sams, Indianapolis, IN, USA.

Skiena, S. S. (2008). **The Algorithm Design Manual,** Second Edition (2nd ed). Springer Science & Business Media.

Teaching and Learning Methods:
This course is mainly theoretical, so the teaching method is based on classroom lectures with active participation by students in problem-solving sessions. In particular, after each new design method is presented students will be challenged to solve problems using it.

Time Distribution and Study Load:
Lectures: 45 hours
Self-study: 9 hours/week

Evaluation Scheme: Exams 100%. Open-book examination is used for both mid-semester and final exam.

Instructor(s): Dr. Chaklam Silpasuwanchai

AT70.03 Theory of Computation, 3(3-0) **Semester: August**

Course Objective: The objective of this course is to provide an exposure to the theory of formal languages, automata and complexity theory.

Learning Outcomes:
The students on the completion of this course would be able to understand the key concepts in the following of theory of computation:

- Finite Automata and Regular Expressions,
- Context-Free Languages and Grammars, Pushdown Automata,
- Turing Machines,
- Computational Complexity Theory.

Pre-requisite: None

Course Outline:

I. Finite Automata and Regular Expressions
   1. Finite State Systems
   2. Non-deterministic Finite Automata
   3. Regular Expressions
   4. The Pumping Lemma for Regular Sets
   5. Closure Properties of Regular Sets
   6. Decision Algorithms for Regular Sets

II. Context-Free Grammars, Pushdown Automata
   1. Derivation Trees
   2. Simplication of Context-Free Grammars
   3. Normal Forms
   4. Pushdown Automata
   5. Properties of Context-Free Languages: The Pumping Lemma for CFL's, Closure Properties of CFL's, Decision Algorithms for CFL's

III. Turing Machines
   1. Computable Languages and Functions
   2. Church's Hypothesis

IV. Undecidability
   1. Properties of Recursive and Recursively Enumerable Languages
   2. Universal Turing Machines and Undecidable Problem
   3. Recursive Function Theory

V. The Chomsky Hierarchy

VI. Computational Complexity Theory: Tractability and Intractability
   1. Deterministic Turing Machines and the Class P
   2. Nondeterministic Turing Machines and the Class NP
   3. Relationship between P and NP
   4. Polynomial Transformation and NP-Completeness
   5. NP-Complete Problems

Laboratory Session(s): None

Learning Resources: Textbooks and reference material.

Textbook:
H.R. Lewis, C.H. Papadimitriou (2015), Element of the Theory of Computation, 2nd edition, Pearson India.

Reference Books:
C. Calude (1988), Theories of Computational Complexity, North Holland.

M. Chandrasekaran, and K.L.P. Mishra (2006), Theory of Computer Science: Automata, Language and Computation, 3rd edition Prentice Hall.

J.E. Hopcroft, J.D. Ullman (2014), Introduction to Automata Theory, Languages, and Computation, International Edition, Addison-Wesley, Massachusetts.

M. Sipser (2012), Introduction to the Theory of Computation, 3rd edition, Pws Pub Co, USA.

C.H. Smith (2012), A Recursive Introduction to the Theory of Computation, Springer Verlag.

R.G. Taylor (1997), Model of Computation and Formal Languages, Oxford University Press.

M.R. Garey and D. S. Johnson (1979), Computer and Intractability, W.H. Freeman and Company, New York.

Teaching and Learning Methods:
The teaching is interactive where abstract concepts are gradually introduced by simple examples to allow students to appreciate the relevant notions and their properties. Relating computational problems to programming languages illuminates both the foundational and practical nature of the theory.

Time Distribution and Study Load:
Lectures: 45 hours.
Tutorials: 20 hours.
Self-study: 110 hours.

Evaluation Scheme:
The final grade will be computed from the following constituent parts:
Mid-term exam: 30%
Final exam: 70%
Open-book examination is used for both mid-semester and final exam.

Instructor(s): Prof. Phan Minh Dũng

AT70.05 Computer Networks, 3(3-0) **Semester: August**

Course Objective: The objective of this course is to provide students with an understanding of network types: wide area networks, local area networks, home networks; circuit switching, packet switching; datagram, virtual circuits; network architecture; error detection, collision avoidance and detection; reliable transmission; the Internet (TCP/IP, routing and addressing, application protocols); ATM networks; network security and quality of Service.

Learning Outcomes :
The students on the completion of this course would be able to:
- Explain the roles of different types of networks and network architectures.
- Develop distributed systems that utilize data link networks, the internet, and ATM networks.
- Design and develop distributed systems keeping network security in mind.
- Design and develop applications utilizing compile networks.

Pre-requisite: None

Course Outline:

I Types of Networks
1. LAN, WAN and Home networks
2. Circuit switching
3. Packet switching
4. Datagram
5. Virtual circuit

II. Network Architecture
1. Layering and Protocols
2. OSI architecture
3. Internet architecture

III. Data Link Networks
1. Framing
2. Error Detection
3. Collision Avoidance
4. Ethernet (802.3)
5. Token Ring (802.5, FDDI)
6. Wireless (802.11)

IV. The Internet
1. TCP/IP Protocol Suite
2. IP Addressing and Routing
3. Domain Name System
4. Application Protocols: SMTP, HTTP, SNMP

V. ATM networks
1. Cell switching
2. Virtual Paths
3. ATM in LAN

VI. Network Security
1. Cryptographic algorithms
2. Security mechanisms
3. Firewalls
4. Denial-of-Service Attack

VII. Applications
1. Multimedia Applications
2. Middleware
3. Web Services

Laboratory Session(s):

Learning Resources: Textbooks and reference books

Textbook:

L.L. Peterson and B.S. David (2000), Computer Networks: A Systems Approach, 2nd edition, Morgan Kaufmam Publishing.

J.F. Kurose and K.W. Ross (2012), Computer Networking – A Top-Down Approach, 6th edition, Pearson.

Reference Books:

D. Comer (2001), Computer Networks And Internets, (CD-ROM by Ralph Droms), 3rd edition, Prentice Hall.

Teaching and Learning Methods:

Mass instruction:	Lectures and regular meeting to discuss on each subjects.
Individual learning:	The materials provided on the web and assigned for self-study.
Group learning:	Make a presentation on assigned topics.

Time Distribution and Study Load:

Lectures:	45 hours
Self-study:	135 hours

Evaluating Scheme:

The final grade will be computed from the following constituent parts:

Project:	30%
Presentations:	30%
Tests and examinations:	40%

Open/Closed-book examination is used for both mid-semester and final exam.

Instructor: Adjunct Faculty

AT70.07 Programming Languages and Compilers, 3 (3-0) Semester: January

Course Objective: The objective of this course is to provide students with an in depth knowledge of concepts that underlie all of the programming languages normally encountered, illustrating those concepts with examples from various languages. Language design and implementation and the ways in which they interact are explored together. Special emphasis in passing and translation.

Learning Outcomes:
The students on the completion of this course would be able to:
- Explain how mechanisms of language design interact.
- Utilize language design mechanisms to implement translators.

Prerequisite: None

Course Outline:
I. Introduction and An Overview
1. Compilation
2. Interpretation

II. Syntax and Passing
1. Regular expressions, and lexical analyzer
2. Context-free grammars
3. Top-down and bottom-up parsers

III. Syntax-Directed Translation and Semantic Analysis
1. Attributes and Semantics Rules
2. Translation scheme
3. Type Checking and Code Generation

IV. Data Types, Names, Scopes and Bindings
1. Type systems, Type Checking and Overloading
2. Static, Dynamic Binding

VI. Control and Data Abstraction
1. Parameter Passing
2. Expression Evaluation
3. Encapsulation
4. Method Binding

VII. Functional and Logic Programming Languages

Laboratory Session(s): None

Learning Resources: Textbooks and reference books

Text Books:
Compilers: Pearson New International Edition: Principles, Techniques, and Tools, Paperback, July 27, 2013

Reference Books:

D. Watt (2004), Programming Language Design Concepts, Wiley, 2004

D. Brown, and D. Watt (2000), Programming Language Processors in Java: Compilers and Interpreters, Morgan Kaufmann Publishers.

R.W. Sebesta (2015), Concepts of Programming Languages, 11th edition, Addison-Wesley.

A. W. Appel, and J. Palsberg (2002), Modern compiler implementation in Java, 2nd edition, Cambridge University Press.

B.J. MacLennan (1999), Principles of Programming Languages: Design, Evaluation, and Implementation, 3rd edition, Oxford University Press.

N. Wirth (1997), Compiler Construction, 1st edition Addison-Wesley.

B. C. Pierce (2002), Types and Programming Languages, The MIT Press.

A. Sampaio (1997), An Algebraic Approach To Compiler Design, World Scientific.

R. Wilhelm, and D. Maurer (1995), Compiler Design Reading, Addison-Wesley.

S.S. Muchnick (2008), Advanced Compiler Design and Implementation, Morgan Kaufmann Publishers.

T. Pittman, J. Peters, and J. Peters (1991), The Art of Compiler Design: Theory and Practice, Prentice Hall.

Teaching and Learning Methods:
The teaching is interactive where abstract concepts are gradually introduced by simple examples to allow students to appreciate the relevant notions and their properties. Assignments and projects will train students on the development of languages and translators.

Time Distribution and Study Load:
Lectures: 45 hours.
Tutorial: 20 hours.
Self-study and Project Workgroup: 110 hours.

Evaluation Scheme:
The final grade will be computed from the following constituent parts:
Mid-semester exam: 20%
Final exam: 50%
Assignments/projects: 30%
Open-book examination is used for both mid-semester and final exam.

Instructor(s): Prof. Phan Minh Dung̃

AT70.09 Computer Graphics and Animation, 3(3-0) **Semester: January**

Course Objective: The objective of this course is to introduce computer graphics as a practical discipline. The underlying theory of computer graphics, as well as implementation algorithms, will be presented in the context of the modern industry-standard 3D graphics programming language OpenGL. Instruction shall be in a laboratory setting with continuous hands-on implementation of concepts and emphasis on creating animated and interactive scenes.

Learning Outcomes:
The students on the completion of this course would be able to :
- Create objects in 3D using OpenGL.
- Create animation sequences using OpenGL.
- Design and implement games and movies.
- Create scenes to specification and animate them.

Pre-requisite: Basic knowledge of two- and three-dimensional co-ordinate geometry and trigonometry and C/C++.

Course Outline:

I. Introduction to a Graphical API (Application Programmer Interface)
   1. Geometric Primitives
   2. Basic Rendering Pipeline
   3. Buffering for Animation
   4. Case Study: OpenGL

II. Curve and Surface Modeling
   1. Approximating Curves by Polygonal Lines
   2. Approximating Curved Surfaces by Triangulated Surfaces
   3. Recursive Subdivision
   4. Case Study: OpenGL

III. Animation
   1. Animation Techniques
   2. Animation by Viewing and Modeling Transformations
   3. Projection Transformations
   4. Matrices and the Mathematics of Affine Transformations
   5. Interaction: Mouse, Keyboard, Menus
   6. Case Study: OpenGL

IV. Color
   1. Color Perception
   2. Color Models and Access Modes
   3. Case Study: OpenGL

V. Illumination
   1. Illumination Models
   2. Shading Models
   3. Case Study: OpenGL

VI. Texture
   1. Texture Objects and Functions
   2. Texture Mapping
   3. Case Study: OpenGL

VII. Advanced Rendering Techniques
   1. Blending
   2. Antialiasing
   3. Fog

VIII. Rendering Algorithms
   1. Scan Converting Lines and Polygons
   2. Line and Polygon Clipping

3. Hidden-Surface Removal: z-buffer

Laboratory Session(s): None

Learning Resources: Textbooks, reference books.

Text Books:

S. Guha, Computer Graphics through Open GL: From Theory to Experiments, 2nd edition.

M. Woo, J. Neider, T. Davis and D. Steiner, OpenGL Programming Guide, 6th edition.

Reference Books:

E. Angel (2000), Interactive Computer Graphics: A Top-Down Approach with OpenGL, Addison-Wesley.

J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes (1996), Computer Graphics Principles and Practice, 2nd edition in C, Addison-Wesley.

D. Hearn and M.P. Baker (2003), Computer Graphics with OpenGL, Addison-Wesley.

D. Shreiner, editor (1999), OpenGL Reference Manual, 3rd edition, Addison-Wesley.

R.S. Wright and M.R. Sweet (1999), OpenGL SuperBible, 2nd edition, Waite Group.

Teaching and Learning Methods:

The teaching method is to combine theory and practice. Lectures will be in a laboratory setting where theoretical expositions will be followed by practical demonstration on the computer; or vice versa where a practical phenomenon will be demonstrated first followed by an explanation. Students will be heavily involved in the instruction process as they will be invited to both ask and answer questions.

Time Distribution and Study Load:

Lectures: 45 hours
Self-study: 135 hours

Evaluation Scheme:

The final grade will be computed from the following constituent parts:
Exams: 50%
Assignments and final project: 50%
Open-book examination is used for both mid-semester and final exam.

Instructor(s): Prof. Sumanta Guha

AT 70.12 Web Application Engineering, 3(2-3) Semester: August

Course Objective: The objective of this course, the students will learn to cope with the challenges of system complexity, massive concurrency, and a … user base by using appropriate technology and a user-centered approach to the design and construction of large-scale Web applications. Students successfully completing Web Application Engineering will be competent database-backed Web application developers capable of designing, deploying, and maintaining large-scale services like Facebook and amazon.com.

Learning Outcomes:
The students on the completion of this course would be able to:
- Build an enterprise-ready Web server using real or virtualized hardware and open source software.
- Develop a concrete project plan for turning an initial concept for a new social Web application into a successful service.
- Design and implement the user model, the content management model, and the user interaction model for a social Web application.
- Improve the user experience of a Web application using best practices for client-side software development.
- Design and implement comprehensive automated unit test and acceptance test suites using modern automated test and continuous integration tools.
- Integrate automated testing into an effective deployment pipeline for a Web application.
- Integrate a Web application into a larger distributed system using best design practices for RESTful or RPC Web services
- Improve the scalability of a Web application through up-front design, bottleneck analysis, and optimization

Prerequisite: Programming experience in C or Java.

Course outline:

I. Web technology background

II. Software architecture for Web applications
1. Layering
2. Model-View-Controller pattern
3. Modern MVC frameworks

III. Data modeling
1. SQL
2. Database normalization
3. Object-relational mapping

IV. Version control

V. Testing
1. Automated unit testing
2. Automated acceptance testing
3. Continuous integration
4. Continuous deployment

VI. Web application security
1. Attack methods
2. Client authentication best practices
3. Cross-site scripting (XSS) and SQL injection attacks
4. Framework support for security

VII. Rich applications
1. Client-side scripting
2. Ajax
3. Rich application components
4. Comet/Websockets

VIII. Web services

1. Representational state transfer (REST)
2. Resource-oriented analysis and design
3. Remote procedure call (RPC) services

IV. Scaling Web applications

1. Bottleneck analysis
2. Scaling strategies

Laboratory Session(s): Installing Linux, Apache, and PostgreSQL; Ruby on Rails; Distributed version control workflow; User authentication, Grails; Ajax; Web sockets; Load balancing; Cloud deployment and performance testing.

Learning Resources: Other references.

Textbooks: None

Others:

- Online documentation for open-source web technology tools.
- Dafydd Stuttard and Marcus Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flows, 2nd Edition, Wiley, 2011.
- Eve Andersson, Philip Greenspun, and Andrew Grumet, Software Engineering for Internet Applications, MIT Press, 2006. Available online at http://philip.greenspun.com/seia/
- Ryan Bates, Rails Casts: Ruby on Rails Screencasts, 2013. Available online at http://railscasts.com.
- David Bryant Copeland, Rails, Angular, Postgres, and Bootstrap: Powerful, Effective, Efficient, Full Stack: Web Development, 2nd Edition, Pragmatic Bookshelf, 2017.
- Brad Dayley, Node.js, Mongo DB, and Angular JS Web Development, Addison-Wesley, 2014.
- Paul M. Duvall, Steve Matyas, and Andrew Glover, Continuous Integration: Improving Software Quality and Reducing Risk, Addison-Wesley, 2007.
- Chad Fowler, Rails Recipes, Rails 3rd Edition, Pragmatic Bookshelf, 2012.
- Martin Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
- Kevin Fu, Emil Sit, Kendra Smith, and Nick Feamster, "Dos and don'ts of client authentication on the Web," In Proceedings of the 10th USENIX Security Symposium, 2001.
- Philip Greenspun, SQL for Web Nerds. Available online: http//philip.greenspun.com/sql/.
- Cal Henderson, Building Scalable Web Sites, O'Reilly, 2006.
- Jez Humble and David Farley, Continuous Delivery: Reliable Software Release through Build, Test and Development, Addison-Wesley, 2010.
- Peter Kim, The Hacker Playbook: Practical Guide to Penetration Testing, Amazon Digital Services, 2014.
- Don Norman, The Design of Everyday Things, Basic Books, 2002.
- Chad Pytel and Tammer Saleh, Rails AntiPatterns: Best Practice Ruby on Rails Refactoring, Addison-Wesley, 2010.
- L. Richardson and S. Ruby, RESTful Web Services, O'Reilly, 2007.
- Sam Ruby, David Bryant Copeland, and Dave Thomas, Agile Web Development with Rails 5.1, Pragmatic Bookshelf, 2018.
- Jim Webber, Savas Parastatidis, and Ian Robinson, REST in Practice: Hypermedia and Systems Architecture, O'Reilly, 2010.

Journals and Magazines: None. [Online resources such as experts' blogs are more important.]

Teaching and Learning Methods:
Teaching methods: AT 70.12 is a project-oriented course in which student teams are paired with client organizations needing online social applications. Using a Web server, programming language, and relational database of their own choice, students take the system from an initial concept through the stages of requirements specification, design, implementation, and usability testing. Along the way, focused laboratory sessions give students experience with specific technologies and techniques useful across many applications,

and lectures introduce students to the best practices and most recent developments in frameworks, middleware, and thick clients.

Time Distribution and Study Load:

Lecture:	2 hrs/week
Lab:	3 hrs/week
Homework:	10 hrs/week over the semester
Self-study:	2 hrs/week

Project (gradually replaces homework): up to 20 hrs/week by the end of the semester

Evaluation Scheme:

The final grade will be completed from the following constituent parts:

Project and lab:	**50%**
Midterm exam:	**20%**
Final exam:	**30%**

Any resources including internet access are allowed during the exams.

Instructor(s): Prof. Matthew N. Dailey

AT70.13 Computer Security, 3(3-0) **Semester: January/Intersem**

Course Objective: The objective of this course is to provide the students with key knowledge about the nature and challenges of computer security, security models, the relationship between policy and security, the role and application of cryptography, especially public key cryptography algorithms and their roles in designing secure protocols, problems and mechanisms in assurance, vulnerability analysis, fire walls, privacy and trust.

Learning Outcomes:
The students on the completion of this course would be able to:
- Analyze security problems
- Use appropriate models and tools to handle security problems.

Prerequisite: None

Course Outline:
I. Introduction
II. Access Control
  1. Security Models and Access Policies
  2. Access Control in Operating Systems
  3. Access Control in Distributed System: Credentials and Certificates, Trust Management, Trust Negotiations.
III. Cryptography and Security Protocols
  1. Conventional Encryption
  2. Public Key Encryption and Digital Signature
  3. Key Exchange and Authentication
  4. Authentication and Key Exchange
  5. Formal Analysis
IV. Systems, Network and Social Networks Security
  1. Electronic Mail Security
  2. IP Security
  3. Web Application Security
  4. Firewalls
  5. Privacy and Trust

Laboratory Sessions: None

Learning Resources: Textbooks, reference books and technical papers.

Text Books:

Lecture Notes
D. Gollman (2014), Computer Security, 3rd edition, John Wiley and Sons Ltd.

Reference Books:
B. Schneier (1996), Applied Cryptography, 2nd edition, John Wiley and Son.

W. Stallings (2016), Cryptography and Network Security: Principles and Practice, 7th edition, Prentice Hall International.

C. P. Pfleeger, and S.L. Pfleeger (2015), Security in Computing, 5th Edition, Prentice Hall.

M. Rhodes-Ousley, B. Rothke, and A. Taylor (2013), Network Security (The Complete Reference), 2nd Edition, McGraw-Hill Osborne Media.

E. D. Zwicky, S. Cooper, and D.B. Chapman (2000), Building Internet Firewalls, 2nd edition, O'Reilly.

C. Adams, and S. Lloyd (2002), Understanding Public-Key Infrastructure: Concepts, Standards and Deployment Considerations, 2nd edition, MacMillan Technical Publishing.

U. O. Pabrai, and V. K. Gurbani (1996), Internet and TCP/IP Network Security, McGraw-Hill.

L. Loeb (1998), Secure Electronic Transactions, Artech House Publishers.

D. O'Mahony, M. Peirce, and H. Tewari (1997), Electronic Payment Systems, Artech House Publishers.

Teaching and Learning Methods:
The teaching is interactive where abstract security models and policies are gradually introduced by simple examples to allow students to appreciate the relevant models and their applications. Assignments and projects will train students on the development of security solutions.

Time Distribution and Study Load:
Lectures: 45 hours
Self-study and Project Workgroup: 125 hours.

Evaluation Scheme:
The final grade will be computed from the following constituent parts:
Final exam: 0%
Assignment and projects: 40%
Open-book examination is used for final exam.

Instructor(s): Prof. Phan Minh Dung

AT70.15 Advanced Topics in Internet Technology, 3(3-0) **Semester: January**

Course Objective: The objective of this course is to provide the students with fundamental issues in network protocol design and implementation and principles underlying TCP/IP protocol design; historical development of the Internet; Internet routing protocols (unicast, multicast and unidirectional); algorithmic issues related to the Internet; multimedia communication (Voice over IP, Real-time protocols); measurement and performance; next generation Internet (IPv6, QoS) and applications.

Learning Outcomes :
The students on the completion of this course would be able to :
- Understand the networking protocols, real-time and multimedia application services,
- Understand the global internet, measurements and metrics, next generation internet and advanced applications

Prerequisite: Computer Networks

Course Outline:

I. Networking Protocols
1. OSI model
2. Internet IP/UDP/TCP
3. Routing in the Internet
4. RIP, OSPF
5. BGP
6. Multicasting
7. Unidirectional Routing

II. Real-time and Multimedia Application Services
1. Real-Time Protocol
2. Voice over IP
3. H.323
4. Audio/VDO Compressions

III. The Global Internet
1. Historical Development
2. IP Address Allocation
3. Domain Name Services

IV. Measurements and Metrics
1. Delay, Latency, Packet Loss, Throughput
2. Link Utilization and Availability
3. Network Monitoring

V. Next Generation Internet and Advanced Applications
1. IPv6
2. MPLS
3. Quality of Services
4. Mobile Internet
5. Grid Computing

Laboratory Session(s): None

Learning Resources: Lecture notes and reference books.

Textbooks: Lecture Notes

Reference Books:

J. Crowcroft, M. Handley, I. Wakeman (1999), Internetworking Multimedia, Morgan Kaufmann.

J. F. Kurose and K. W. Ross (2012), Computer Networking - A Top-Down, 6th edition, Pearson.

L. L. Peterson and B. S. Davie (2000), Computer Networks: A Systems Approach, 2nd edition, Morgan Kaufmann.

S. A. Thomas (1995), IPng and the TCP/IP Protocols, Wiley.

W. R. Stevens (1994), TCP/IP Illustrated, vol. 1, Addison-Wesley.

O. Hersent, D. Gurle, J.-P. Petit (2000), IP Telephony, Addison-Wesley.

Teaching and Learning Methods:
The teaching and learning methods may include lectures, classroom exercises and presentations, home assignments, field visits, group projects, etc.

Time Distribution and Study Load:
Lectures: 45 hours
Self-study: 135 hours

Evaluation Scheme:
The final grade will be computed from the following constituent parts:

Project:	30%
Presentations:	30%
Tests and examinations:	40%

Open/Closed-book examination is used for both mid-semester and final exam.

Instructor(s): Dr. Adisorn Lertsinsrubtavee

AT70.16 Computational Geometry and Applications, 3(3-0) Semester: January

Course Objective: The objective of this course is to provide students with an introduction to both the theory and applications of the discipline of computational geometry which is concerned with the solving of computational problems arising from geometric questions. Essential theory and algorithms will be covered and content will be motivated by practical problems. Implementations of geometric algorithms in a high-level language will be covered. Course will be seminar-style.

Learning Outcomes:
The students on the completion of this course would be able to:

- Apple a repertoire of standard techniques for design of geometric algorithms as well as the related data structures.
- Analyze a geometric problem and devise an algorithm to solve it.
- Understand the particular design requirements of different geometric problem domains.
- Analyze problems in related areas, e.g., computer graphics, to understand and resolve the geometric issues.
- Given a geometric algorithm, be able to analyze its time and space complexity.

Prerequisite: Data Structures and Algorithms, or Instructor Consent.

Course Outline:

I. Convex Hulls
   5. Properties
   6. Algorithms
   7. Handling Degeneracy and Robustness
   8. Applications Domains

II. Line Segment Intersections
   7. Algorithms
   8. Plane Sweep Method
   9. Doubly-Connected Edge List
   10. Application Domains

III. Polygon Triangulation
   1. Properties
   4. Algorithms
   5. Applications Domains

IV. Linear Programming
   4. Geometric View of Linear Programming
   5. Two-variable Linear Programming: Intersecting Half-Planes
   6. Incremental Linear Programming
   7. Randomized Linear Programming: Backward Analysis
   8. Linear Programming in Higher Dimensions

V. Voronoi Diagrams
   1. Properties
   2. Algorithms
   4. Application Domains

VI. Delaunay Triangulation
   4. Point Set Triangulation
   5. Properties: Delaunay Triangulation as Dual of the Voronoi Diagram
   3. Algorithms
   5. Application Domains

VII. Point Location
   1. Trapezoidal Maps
   2. Randomized Incremental Algorithms

VIII. Robot Motion Planning
   1. Work space and Configuration Space

2. Point Robots
6. Motion Planning Techniques

IX. Binary Space Partition
1. Properties
2. Painter's Algorithm
3. Construction

X. Special Topics
According to interest.

Laboratory Session(s): None

Learning Resources: Textbooks, reference books, journals.

Textbook:

M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf (2000), Computational Geometry: Algorithms and Applications, 2nd edition, Springer Verlag.

Reference Books:

J-D. Boissonnat and M. Yvinec (1998), Algorithmic Geometry, Cambridge University Press.

K. Mulmuley (1998), Computational Geoemetry: An Introduction through Randomized Algorithms, Prentice Hall.

J. O'Rourke (1998), Computational Geometry in C, 2nd edition, Cambridge University Press.

F. P. Preparata and M. I. Shamos (1991), Computational Geometry: An Introduction, Springer-Verlag.

Journals and Magazines:
- International Journal of Computational Geometry and Applications, World Scientific.
- Computational Geometry: Theory and Applications, Elsevier.

Teaching and Learning Methods:
This course is mainly theoretical so the teaching method is based on classroom lectures with active participation by students in problem-solving sessions. In particular, after each new design method is presented students will be challenged to solve problems using it.

Time Distribution and Study Load:
Lectures: 45 hours
Self-study: 135 hours

Evaluation Scheme:
The final grade will be computed from the following constituent parts:
Examination 100%
Open-book examination is used for both mid-semester and final exam.

Instructor(s): Prof. Sumanta Guha

AT70.18 Software Architecture Design, 3(3-0) Semester: January

<u>Course Objective</u>: The objective of this course is to provide the students with software architects building complex systems must create the illusion of simplicity through decomposition, abstraction, and encapsulation of functionality. In Sotware Architecture Design, we focus primarily on architecture in the large, at the level of the enterprise, in which multiple applications must work together to support or automate business processes through concurrent access to large amounts of complex persistent data. We study the design and implementation of large-scale enterprise information system infrastructures, enterprise application integration, and the modern asynchronous distributed systems middleware technologies necessary to support enterprise application development. Students will learn industry best practices through the study of architectural design patterns and put the principles to practice they learn by designing and constructing an architectural prototype for a significant real-world software project.

<u>Learning Outcomes</u>:
The students on the completion of this course would be able to :

- Identify the most appropriate architectural design patterns that should be used to meet functional and non-functional requirements for an enterprise application development project;
- Implement an architectural design pattern in the context of a particular enterprise application development framework;
- Identify the most appropriate architectural design patterns for distributed systems that should be used to meet functional and non-functional requirements for an enterprise application integration project;
- Implement a distributed systems design pattern in the context of a particular enterprise application integration framework;
- Design a distributed system by decomposing a business process into services and integrating process logic;
- Automate a business process by implementing services and integrating process logic in the context of a particular business process management framework.

<u>Prerequisites</u>: Knowledge of object-oriented analysis and design; programming experience with an object-oriented programming language such as C++, Java, Ruby, Python, or C#.

<u>Course Outline</u>:

I. <u>Introduction</u>

II. <u>Enterprise software architectural design patterns</u>
   1. Domain logic
   2. Mapping to relational databases
   3. Concurrency
   4. Session state
   5. Distributed objects
   6. Implementation case studies

III. <u>Enterprise application integration patterns</u>
   1. Messages and message channels
   2. Routing
   3. Transformation
   4. Endpoints
   5. Message brokers and process brokers
   6. System management
   7. Implementation case studies

IV. <u>Architectural patterns for service-based systems</u>
   1. Business process patterns
   2. Business service analysis and design
   3. Technical patterns
   4. Implementation case studies

Laboratory Session(s):
Although there is no formal lab for this course, there will be several practical tutorials on the implementation of the architectural patterns discussed in class on enterprise application development and business process management frameworks such as Java EE, Spring, OSGi, Apache Camel, and Activity.

Learning Resources: Textbooks, reference books, journals and magazines.

Textbooks:
Fowler (2002), Patterns of Enterprise Application Architecture, Addison-Wesley.

Hohpe and Woolf (2004), Enterprise Integration Patterns, Addison-Wesley.

Erl, T. (2009), SOA Design Patterns, Prentice Hall.

Daigneau, R. (2012), Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services, Addison-Wesley.

Reference books:
Bass, Clements, and Kazman (2003), Software Architecture in Practice, 2nd edition, Addison-Wesley.

Larman, C. (2005), Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edition, Prentice Hall.

Buschmann, Henney, and Schmidt (2007), Pattern-Oriented Software Architecture Vol.4: A Pattern Language for Distributed Computing, Wiley.

Panda, D., Rahman, R., Cuprak, R., and Remijan, M. (2013), EJB 3 in Action, 2nd edition, Manning.

Rademakers, T., and Dirksen, J. (2009), Open Source ESBs in Action, Manning.

Rademakers, T., (2012), Activiti in Action, Manning.

Walls, C., and Breidenbach, R. (2011), Spring in Action, 3rd edition, Manning.

Journals and Magazines: None. [Online resources such as experts' blogs are more important.]

Teaching and Learning Methods:
Lectures introducing the fundamental architectural design patterns for enterprise applications, enterprise application integration, and business process management will be accompanied by in-class discussion, in-class exercises, in-class technical tutorials that students should follow and replicate, homework assignments on the implementation of patterns discussed in lecture, and an extended project.

Time Distribution and Study Load:
Lectures: At least 32 hours
Tutorials: At least 10 hours
Self-study: At least 9 hours per week
Evaluation Scheme:
The final grade will be computed from the following constituent parts:
Homework: 20%
Midterm-exam: 25%
Project: 25%
Final-exam: 30%
Any resources including Internet access are allowed during the exams.

Instructor(s): Prof. Matthew N. Dailey

AT70.19 Software Development and Quality Improvement, 3(3-0) Semester: January

Course Objective: The objective of this course is to provide the students with designing, developing, and improving complex software systems requires a mastery of analytical and technical skills, as well as a knowledge of appropriate processes, architectures and design patterns. This course teaches the fundamental skills of software engineering, drawn from research and best-practice on large open source and commercial software projects. Students will learn techniques and tools for modeling, analyzing, developing and evaluating complex software systems. The emphasis will be on rapid implementation of complex systems through agile development processes, visual development tools, software frameworks, and integration of open source and commercial components.

The course will also improve students' practical software engineering skills by having them plan and execute a significant software development project. Students may make a specific contribution to an existing large open source project or start a new project of their own choice. Projects with the potential to play a role in development of the Asian region will be strongly encouraged.

Learning Outcomes :
The students on the completion of this course will be able to:

- Understand the specificity of software engineering compared with other engineering fields.
- Engage in a complex software development project.
- Utilize the skills developed in the class to manage a project with the necessary autonomy.

Prerequisite: AT 70.18 (Software Architecture Design)

Course outline:
Software Development Process

1. Software Engineering Concept
2. Modeling with UML
3. Project Organization and Communication
4. Requirements Elicitation
5. Analysis and Design

Software Life Cycle

1. Lifecycle phases and process
2. Software Life cycle models
3. Waterfall, Rapid Prototyping, Agile
4. Unified Process

Tools and Methods

1. Programming methodologies
2. CASE Tools
3. Build control, version control, and integration
4. Visual analysis tools

Software Configuration Management

1. Monitoring and auditing
2. Build and Release management
3. Continuous Integration

Testing and Quality

1. Software Testing fundamentals and techniques
2. Debugging
3. Software Quality assurance
4. Verification and validation
5. Software Improvement
6. Dynamic analysis
7. Issue tracking

Laboratory Session(s): None

Learning Resources: Textbooks and reference books.

Textbook:

Lecture notes provided by instructor

Reference Books:

B. Bruegge and A. Dutoit (2014), Object-Oriented Software Engineering Using UML, Patterns, and Java™, 3rd edition. Addison-Wesley, ISBN 0-13-606125-7. (Recommended)

P. Stevens (2006), Using UML: Software Engineering with Objects and Components, 2nd edition. Addison-Wesley, ISBN 0-321-26967-5.

I. Sommerville (2005), Software Engineering, 7th edition, Addison-Wesley, ISBN 0-321-21026-3. (Recommended)

C. Larman (2005), Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice-Hall.

R. S. Pressman (2004), Software Engineering: A Practitioner's Approach, 6th edition. McGraw-Hill.

B. Bruegge and A. H.Dutoit (2004), Object-Oriented Software Engineering: Using UML, Patterns, and Java, 2nd edition, Prentice-Hall, ISBN 0-13-1911791.

Fowler (2003), UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd edition, Addison-Wesley.

Brown, Malveau, McCormick and Mowbray (1998), AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis, Wiley.

F. P. Brooks (1995), The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, ISBN 0-201-83595-9.

E. Gamma, R. Helm, R. Johnson and J. Vlissides (1995), Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995. ISBN 0201633612.

Teaching and Learning Methods:
The courses is based on lectures and practicals. It is an interactive learning process where the students are highly encouraged to participate not only to the proposed activities but also to decide what are the subjects they want to focus more on. Students have to realize a prototype so they concretely apply what they are learning. They have to prepare a short talk to show they have understood some fundamental concepts.

Time Distribution and Study Load:
Lecture hours: 45
Project is mainly realize as a homework.

Evaluation Scheme:
The final grade will be computed from the following constituent parts:
Homework: 30%
Project: 30%
Final exam: 40%
Open book is used for final exam.

Instructor(s): Dr. Apichon Witayangkurn

AT70.22 Big Data Analytics, 3(2-3) **Semester: January**

Course Objectives: Big data is the term applied to datasets whose size or type are beyond the capacity of traditional data storage and processing systems to handle quickly. Modern high-performance computing infrastructure and algorithms enable the capture and analysis of many kinds of big data from various sources such as transactions, people, devices, sensors, and vehicles. Enterprises in every sector can improve efficiency and overcome challenges through appropriate application of big data analytics. This course will provide students with fundamental knowledge on large-scale storage and analytic techniques, preparing them to create state-of-the-art solutions for real-world challenges. The course will introduce state of the art frameworks such as Hadoop, Spark, and other tools in the Hadoop ecosystem. In a final project, students will apply the knowledge they gain to an application domain of their own interest.

Learning Outcomes:

Students, on completion of the course, would be able to:

- Apply fundamental concepts of big data and large-scale data processing.
- Set up a computing environment for big data analytics, including hardware, distributed systems frameworks, and analytic tools.
- Design a data processing stream for a big data platform.
- Develop and deploy an analytic model to a big data platform.
- Implement programs that turn big data into insights that deliver value in selected domains.

Prerequisite(s): None

Course Outline:

I. Introduction
1. Overview and History of Big Data
2. Big Data Architecture and Data Lake
3. Real-world applications and use cases

II. Big Data Processing
1. Historical Data Processing Technologies
2. Concepts of Data Processing System
3. Data Pre-Processing, Validation, Filtering, Outlier Detection and Removal
4. Batch vs. Real-time processing

III. Big Data Platforms
1. The Evolution of Hadoop
2. Hadoop Architecture and its Key Principles
3. Hadoop Ecosystem
4. Real-time Analytics Platform with Kafka and Spark

IV. Big Data Analytics Techniques
1. Classification, Predictive Modeling, Time Series, Clustering
2. Distributed and Scalable Machine Learning with Apache Mahout
3. Large-Scale Data Summarization, Query and Analysis with Apache Hive

V. Big Data Visualization
1. Overview of Data Visualization
2. Challenges and Techniques
3. Visualization tools for Big Data

Laboratory Session(s):
1. Tutorials on installation of the Hadoop system.
2. Tutorials on MapReduce.
3. Tutorials on data processing on the Hadoop system.
4. Tutorials on Apache Hive.
5. Tutorials on Apache Spark.
6. Tutorials on big data visualization tools.

Learning Resources: References books.

Textbook(s): No designated textbook, but class notes and handouts will be provided

Reference Books:

Venkat Ankam (2016), Big Data Analytics, Packt.

Nathan Marz, and James Warren (2015), Big Data: Principles and Best Practices of Scalable Realtime Data Systems, Manning.

Arshdeep Bahga, and Vijay Madisetti (2016), Big Data Science & Analytics: A Hands-On Approach.

Journals and Magazines:

- IEEE Transactions on Big Data
- SIGKDD EXPLORATIONS
- Pervasive and Mobile Computing – Elsevier
- Pervasive Computing – IEEE

Teaching and Learning Methods:

1. **Lectures and class discussion**: Students will receive the lecture notes and the weekly lecture schedule at the beginning of the course, and they will be requested to read the lecture notes before coming to class.
2. **Laboratory sessions**: Laboratory sessions will reinforce lectures. Software and example data will be provided to establish a consistent environment. The lab instructor will provide basic guidelines to familiarize students with the lab's objectives. Students must complete each lab and submit the homework assignments.

Time Distribution and Study Load:

Lecture:	30 hours.
Laboratory:	45 hours.
Self-study:	45 hours.

Evaluation Scheme:

The final grade will be computed from the following constituent parts:

Laboratory reports and exercises:	30%
Project:	40%
Final exam:	30%

Exams are open book.

Instructor(s): Dr. Apichon Witayankurn

AT71.03 E-Business Development and Technology, 3(3-0) **Semester: August**

Course Objective: The objective of this course is to provide students with an overview of the key concepts, strategies, business models, and technologies behind E-business/E-commerce. The course will address the opportunities and challenges of doing business on the Internet, and the challenges of introducing e-business techniques into existing organizations. Essential global issues related to E-Business and use of social media for business/marketing purpose will be covered.

Learning Outcomes : The students on the completion of this course would be able to

- Setup an online business using internet, web technologies, and social media
- Analyze online business strategies, all E-commerce business models, E-commerce marketing and advertising.
- Apply online security and payment into websites of online business.

Prerequisite: None

Course Outline:

I. Introduction to Electronic Commerce
   1. Electronic Business Terms
   2. Benefits and Limitations
   3. Driving Forces and Impacts

II. E-Commerce Retaling and Services
   1. Online Retail Sector
   2. Internet Shopping
   3. Online Purchase Decision Aids
   4. Online Service Sector

III. E-Commerce Marketing and Advertising
   1. Consumer Behavior Model
   2. Advertisement Methods and Strategies
   3. Internet Marketing Technologies
   4. Online Marketing Communications

IV. E-Commerce Business Models
   1. Online Business Models
   2. B2C Business Models
   3. B2B Business Models

V. Online Media
   1. Online Content
   2. Online Publishing Industry
   3. Online Entertainment Industry

VI. Business-to-Business (B2B)
   1. B2B Models
   2. Sell-side Marketplaces
   3. Sell-side Intermediaries
   4. Buy Side: e-Procurement
   5. Collaborative Commerce
   6. E-Marketplaces and B2B Exchanges

VII. Social Networks and Communities
   1. Social Commerce
   2. Online Auction
   3. E-Commerce Portals
   4. Social Marketing

VIII. E-Commerce Security and Payment Systems
   1. E-Commerce Security Environment
   2. Threats and Technology Solutions

3. E-Commerce Payment Systems
4. Electronic Billing Presentation and Payment

IX. Infrastructure for E-Commerce
1. Internet, Intranet, Extranet
2. The Web
3. Mobile Platform: M-Commerce
4. Websites

X. Global Issues
1. International Issues
2. Cultural Issues
3. Legal Issues
4. Taxation

Laboratory Session(s): None

Learning Resources: Textbooks, Reference Books, Journals and Articles.

Textbooks:

K. C. Laudon and C. G. Traver (2014), E-Commerce, Business, Technology, and Society, Pearson.

Reference Books:

G. P. Schneider (2011), E-Business, 9th edition, Course Technology.

E. Turban, J. Lee, D. King, J. McKay, and P. Marshall (2012), Electronic Commerce, 7th edition, Prentice Hall.

Journals and Articles:

- Electronic Commerce Research and Application Journal
- Electronic Markets
- International Journal of Electronic Business
- International Journal of Electronic Commerce
- International Journal of Electronic Finance
- Journal of Electronic Commerce in Organizations
- Journal of Organizational Computing and Electronic Commerce
- Quarterly Journal of Electronic Commerce

Teaching and Learning Methods:

Multiple participant-centered teaching methods focusing on participative learning are used which include lectures, reading discussions and presentations, home assignment, and a group project.

Time Distribution and Study Load:

Two hours of lectures, and one hour of readings discussion/presentation per week.
Extra time is needed for assignment and course project.

Evaluation Scheme:

The final grade will be computed from the following constituent parts:
Project (30%).
Mid-semester exam (25%),

Final exam (20%),
Reading presentation (10%),
Assignment (10%)
Closed-book examination is used for both mid-semester and final exam.

Instructor: Dr. Vatcharaporn Esichaikul

AT71.05 Information Systems Development and Management, 3(3-0) Semester: January

Course Objective: Nowadays, organizations and businesses employ different types of information systems (IS) to process their day-to-day operations, and to support their operational, tactical and strategic decisions. This course provides an overview of fundamental information systems development and management. It examines important software development process and methods to analyze, model and design an information system of an organization. Students will be exposed to a blend of traditional and emergent techniques to grasp a thorough understanding of key software development methods and techniques, which are appropriate for different types of systems. In addition, key software project management principles (planning, organizing, staffing, directing, and controlling) are discussed and practiced throughout.

Learning Outcomes: The students on the completion of this course would be able to:

- understand the concepts of information systems development life cycle, software development process and methods, software project management, system analysis and design, interface design, system implementation and maintenance;
- Apply software development methodologies and techniques to real-life projects;
- Manage and plan an information system project;
- Prepare and present project proposal;
- Determine and model system requirements;
- Communicate ideas orally, in written form and through information technologies.

Prerequisite: AT71.01 Database Design

Course Outline:

I Introduction to Information Systems Development and Management
1. Roles and Importance of Information Systems
2. Introduction to Software Engineering
3. Origins of Software
    - Outsourcing
    - Sources of Software
    - COTS
    - Reuse

II System Development Life Cycle and Software Process
1. Relational Model Concepts
    - Waterfall: Traditional Waterfall and V-Model
    - Incremental
    - Evolutionary: Prototyping and Spiral
    - Rational Unified Process
2. Process Activities
    - Specification
    - Development
    - Validation
    - Evolution

III Agile Development
1. Agile Values and Principles
2. Plan-driven and Agile Development
3. Agile Development Methodologies:

IV Managing Information System Projects
1. Managing Information System Project
2. Representing and Scheduling Project Plan
3. Risk Management
4. People Management and Teamwork
5. Project Management Software

V Identifying and Selecting Systems Development Projects

1. Identifying and Selecting Systems Development Projects
2. Corporate and Information Systems Planning

VI Initiation and Planning System Development Project
1. Assessing Project Feasibility
2. Cost/Benefit Analysis and ROI
3. Building a Baseline Project Plan

VII Requirements Determination and Analysis
1. Functional and Non-Functional Requirements
2. Requirements Specification
3. Requirements Engineering Processes
    - Requirements elicitation and analysis
    - Requirements validation
    - Requirements management

VIII Requirements Modeling
1. Context Model
2. Interaction Model
3. Structural Model
4. Behavioral Model
5. CASE Tools

IX Designing the Interface
1. Forms and Reports
2. Interface Design Techniques

X Implementation
1. Coding
2. Verification and Validation
3. Testing
4. Installing
5. Documentation
6. User Training

XI Maintenance
1. Types of Maintenance
2. Cost of Maintenance
3. Managing Maintenance

Laboratory Session(s): None

Learning Resources: Textbooks, references books, journals and magazines and others.

Textbooks:

J.S. Valacich and J.F. George (2017), Modern Systems Analysis and Design, 8th edition, Addison Wesley.

I. Sommerville (2015), Software Engineering, 10th edition, Addison Wesley.

Reference Books:

R. Pressman (2009), Software Engineering: A Practitioner's Approach, 7th edition, McGraw-Hill.

C. Larman (2005), Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edition, Pearson.

M. Fowler (2004), UML Distilled: A Brief Guide to the Standard Object Modeling language, 3rd edition, Addison Wesley.

K.C. Laudon and J.P. Laudon (2014), Management Information Systems: Managing the Digital Firm, 13th edition, Pearson.

Journal:

- IEEE Transactions on Software Engineering (TSE), IEEE
- Information and Software Technology, Elsevier, ISSN: 0950-5849
- Requirements Engineering, Springer, ISSN: 0947-3602 (Print) 1432-010X (Online)

Teaching and Learning Methods:

Project-oriented learning and multiple participatory learning methods, focusing on participative learning are used, which include lectures, in-class exercises, discussions and presentation, case studies, assignments and group projects.

Time Distribution and Study Load:

For each week, two hours of lectures, and one hour of in-class exercises, games, case-studies, discussions and presentations per week. Extra time is needed for assignments and course projects.
Lectures: 45 hours
Self-study: 135 hours

Evaluation Scheme:

The final grade will be computed from the following constituent parts:
Mid-semester exam (25%),
Final exam (25%),
Assignments & Presentation (25%),
Project (25%).
Open-book examination is used for both mid-semester and final exam.

Instructor(s): Dr. Chutiporn Anutariya

AT71.07 Information Retrieval and Data Mining, 3(3-0) **Semester: August**

Course Objective: The objective of this course is to introduce information retrieval and data mining techniques with a view to practical application. Topics covered will include association and rule generation, classification and prediction, cluster analysis, data stream mining, social network analysis, Boolean retrieval, index construction and compression, vector space model, relevance feedback and query expansion, probabilistic information retrieval. Practical case studies will use both commercial and non-commercial software package.

Learning Outcomes: The students on the completion of this course would be able to:

- Analyze a collection of data collected by a business or organization from the point of data mining.
- Organize data archived by a business or organization for the purpose of efficient search and retrieval.
- Apply the fundamental issues and algorithms in data mining and information retrieval.
- Adapt existing methodology to rapidly evolving technology for information storage and representation.

Prerequisite: None

Course Outline:

I Boolean Retrieval
1. Inverted index
2. Processing Boolean queries
3. Extended Boolean model
4. Ranked retrieval

II Index Construction
1. Blocked sort-based indexing
2. Single-pass in-memory indexing
3. Distributed indexing
4. Dynamic indexing

III Index Compression
1. Statistical properties of terms in information retrieval
2. Dictionary compression
3. Postings file compression

IV Scoring and the Vector Space Model
1. Parametric and zone indexes
2. Vector space model for scoring

V Mining Frequent Patterns, Associations, And Correlations
1. Efficient and scalable frequent itemset mining methods
2. Mining association rules
3. Association mining to correlation analysis
4. Constraint-based association mining

VI Classification And Prediction
1. Classification and prediction methods
2. Accuracy and error measures
3. Evaluation techniques
4. Model selection

VII Cluster Analysis
1. Clustering methods
2. High-dimensional data
3. Constraint-based cluster analysis
4. Outlier analysis

VIII Special Applications
1. Mining data streams
2. Mining time series data
3. Graph mining

4. Social network analysis

IV Case studies

Laboratory Session(s): None

Learning Resource: Textbooks and others

Textbook:

J. Han and M. Kamber (2006), Data Mining: Concepts and Techniques, 2nd edition, Morgan Kaufmann.

C. D. Manning, P. Raghavan, and H. Schutze (2009), An Introduction to Information Retrieval, Cambridge University Press.

Others:

M. J. A. Berry and G. Linoff (1997), Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management, Wiley.

I. H. Witten and E. Frank (2001), Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann.

T. Soukup and I. Davidson (2002), Visual Data Mining: Techniques and Tools for Data Visualization and Mining, Wiley.

P. Tan, M. Steinbach and V. Kumar (2005), Introduction to Data Mining, Addison-Wesley.

D. T. Larose (2006), Data Mining Methods and Models, Wiley.

Teaching and Learning Methods:

This course is mainly theoretical so the teaching method is based on classroom lectures with active participation by students in problem-solving sessions. Mining and retrieval method will be demonstrated as well with help of practical software.

Time Distribution and Study Load:

Lectures: 45 hours
Self-study: 135 hours

Evaluation Scheme:

Exams 100%.
Open-book examination is used for both mid-semester and final exam.

Instructor(s): Adjunct Faculty

AT71.08 Data Modeling and Management, 3(2-3) Semester: August

Objective: The course emphasizes on emerging data models and technologies suitable for managing different types and characteristics of data. Student will develop skills for analyzing, evaluating, modeling and developing database applications with concerns on both technical and business requirements.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Explain data modeling and management concepts.
2. Design and organize various types of data using a relational and non-relational data models.
3. Analyze the characteristics and requirements of data and select an appropriate data model.
4. Identify, implement and perform frequent data operations (CRUD: create, read, update and delete) on relational and NoSQL databases.
5. Describe the concepts and the importance of big data, data security, privacy and governance.
6. Describe the concepts and the importance of data engineering and data visualization.

Prerequisites: None

Course Outline:

I. Recall: Relational Data Model and Management
   1. Relational Model Concepts
   2. SQL
   3. Relational Database Design and Normalization
   4. Relational Database Management Systems (RDBMSs)

II. NoSQL Data Modeling and Management
   1. NoSQL Concepts and Characteristics
   2. Major Categories of NoSQL Data Models
   3. NoSQL Database Design
   4. NoSQL Features and Operations

III. Data Distribution
   1. Data Sharding and Replication Models
   2. CAP Theorem

IV. Transaction Processing and Consistency Models
   1. Transaction Processing Concepts
   2. ACID Model
   3. BASE Model

V. Large Scale Data Handling
   1. Big Data characteristics
   2. Big Data Modeling and Management

VI. Applications and Case Studies

VII. Data Engineering
   1. Business Understanding
   2. Data Acquisition and Understanding
   3. Data Cleansing
   4. Data Preparation, Transformation and Feature Engineering

VIII. Introduction to Related Topics
   1. Data Security
   2. Data Privacy and Legal Issues,
   3. Data Governance: Social and Ethical Issues, Biasness (gender, religions, etc.)

Laboratory Session(s):

30 hours of laboratory sessions of NoSQL data stores, tools, CRUD operations, and API development/usage.

Learning Resources:

Textbooks:

A. Meier and M. Kaufmann: SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management, Springer, 2019, ISBN 978-3658245481

Reference Books:

M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, O'Reilly, 2017, ISBN 978-1449373320

D. Sullivan, NoSQL for Mere Mortals, Addison-Wesley, 2015, ISBN 978-0-1340-2321-2

P. Sadalage and M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2013, ISBN 978-0-3218-2662-6

E. Redmond and J. R. Wilson, Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement, 2012, ISBN 978-1-93435-692-0

G. Harrison,Next Generation Databases: NoSQL and Big Data, Apress, 2015, ISBN 978-1-4842-1329-2

I. Robinson, J. Webber and E. Eifrem, Graph Databases: New Opportunities for Connected Data, 2/E, O'Reilly, 2015, ISBN 978-1-491-93200-1

R. Elmasri and S. Navathe: Fundamentals of Database Systems, 7/E, Addison-Wesley, 2015

Journals and Magazines:

IEEE Transactions on Knowledge and Data Engineering
ACM Trans. Database Systems.
ACM Trans. Information Systems.

Teaching and Learning Methods:

1. Lecture and Lab
2. Assignment
3. Project
4. Discussions and Case Studies

Evaluation Components

Mid-semester Exam: 25%
Lab & Assignments: 20%
Project: 30%
Final Exam: 25%

Instructor(s): Dr. Chutiporn Anutariya

AT71.09 Business Intelligence and Analytics, 3(3-0) **Semester: January**

<u>Objective:</u> Business intelligence (BI) is a process of analyzing business data to obtain business insights and actionable intelligence and knowledge, in order to support better business decision making and capture new business opportunities. This course will give students an understanding of the principles and practices of business intelligence and data analytics to support organizations in conducting their business in a competitive environment.

<u>Learning Outcomes:</u> Students, on successful completion of the course, will be able to

1. Explain the concepts characteristics of BI and data analytics
2. Describe multiple business problem/decision making domains requiring BI and data analytics
3. Apply BI and data analytic tools and technologies to develop BI applications
2. Integrate BI applications with other information systems as part of a business process
3. Define a BI strategy for an organization
4. Manage a BI project for an organization
5. Describe big data analytics and applications

<u>Prerequisites:</u> Data Modeling Management

<u>Course Outline:</u>

I. <u>Introduction to Business Intelligence</u>
  1. BI Definition
  2. BI Concepts
  3. Business Intelligence, Analytics, and Data Science
  4. Business Intelligence to Support Decisions

II. <u>Data Warehousing for BI</u>
  1. DW design
  2. Multidimensional data modelling and analysis
  3. ETL process

III. <u>Categories of Data analytics:</u>
  1. Descriptive Analytics
  2. Predictive Analytics
  3. Prescriptive Analytics

IV. <u>Descriptive Analytics</u>
  1. Descriptive Statistics
  2. Business Performance Management
  3. Data Visualization and Dashboard Design

V. <u>Predictive Analytics</u>
  1. Data Mining (Text Analytics and Text Mining, Web Analytics, Web Mining, and Social Analytics)
  2. Predictive Modeling

VI. <u>Overview of Prescriptive Analytics</u>
  1. Optimization
  2. Multi-Criteria Systems

VII. <u>Technical Aspects</u>
  1. BI Architecture
  2. BI Tools and Technologies

VIII. <u>BI Applications</u>
  1. BI Maturity
  2. BI Strategies
  3. BI Project (case study)

IX. <u>Overview of Big Data</u>
  1. Big Data Analytics

2. Example of Big Data Applications

Laboratory Session(s): None

Learning Resources:

Indicate Textbooks, References, Journals and Magazines and Others

Text Book:

Business intelligence, analytics, and data science, by Ramesh Sharda;Dursun Delen; Efraim Turban, Pearson Publisher, 2018

Recommended books:

1. Business Analytics (2nd Ed.) by James Evans, Pearson, 2017.
2. Business Analysis for Business Intelligence (1st Ed) by Bert Brijs, Auerbach
3. Publications, 2013.
4. Business Intelligence Guidebook (1st Ed) by Rick Sherman, Morgan Kaufmann, 2014.
5. Fundamentals of Business Intelligence by Wilfried Grossmann and Stefanie Rinderle- Ma, Springer, 2015.

Journals:

- Decision Support Systems
- International Journal of Business Intelligence and Data Mining
- International Journal of Business Intelligence Research
- Journal of Big Data

Teaching and Learning Methods:

1. Lecture
2. Assignment
3. Course Project
4. Real-world case studies
2. Self-learning

Evaluation Components

1. Exam: 40%
2. Assignment: 20%
3. Course Project: 40%

Instructor(s): Dr. Vatcharaporn Esichaikul

AT71.9002 Selected Topics: Seminar in Electronic Government, 3(3-0) Semester: Inter-Sem

Course Objective:

The objective of this course is to provide the students with an understanding of concepts and principles of e-governments. Insights of e-government developments and challenges are discussed. All technical, managerial, and social aspects of e-government are addressed. The course is a mixture of lectures on fundamentals, student presentations of research from the academic journals, and a study report on seleted e-government topics .

Learning Outcomes :

The students on the completion of this course would be able to :

- Identify a good potential e-government project for development
- Apply the design methodology and techniques for implementing e-government
- Apply e-government framework and models
- Evaluate the performance of an e-government project.

Prerequisite: None

Course Outline:

I. Understanding e-Government

1. e-Government defined
2. Maturity models

II. E-government services

1. G2B
2. G2C
3. G2G
4. C2C

III. E-government Development

4. Phases
5. Critical Success Factors
6. Challenges and Barriers

IV. E-participation

4. E-democracy
5. E-voting
6. Citizen portal

V. Technologies underlying e-government

1. Web 2.0
2. XML
3. Open source software
4. Metadata standard

VI. Mobile government

VII. Other Issues

1. E-government planning
2. Digital divide
3. Cultural issues
4. Social value

VIII. Global case studies of e-government

IX. Student presentations of research papers

Laboratory Session(s): None

Learning Resources: Reference Books, Journals and Magazines.

Reference Books:

- Beyond e-Government & e-Democracy: A Global Perspective, Shark and Toporkoff, Public Technology Institute & items International, 2008.
- Digital Government, West, Princeton University Press, 2007.
- E-Government: From Vision to Implementation, Bhatnagar, SAGE Publications, 2008.

Journals and Magazines

- Electronic Government, an International Journal (EG)
- International Journal of Electronic Democracy (IJED)
- International Journal of Electronic Governance (IJEG)
- Electronic Journal of e-Government

Teaching and Learning Methods:
Multiple participant-centered teaching methods focusing on participative learning are used, which include lectures, reading discussions and presentation, home assignments and group project.

Time Distribution and Study Load:
Two hours of lectures, and one hour of readings discussion/presentation per week.
Extra time is needed for assignment and course project.

Evaluation Scheme:
The final grade will be computed from the following constituent parts:

In-class presentations:	20%
Assignment:	20%
Study report:	35%
Final exam:	25%

Instructor(s): Dr. Vatcharaporn Esichaikul

AT82.01 Computer Programming for Data Science And Artificial Intelligence, 2(0-6) **Semester: August**

Objective: The course objective is to provide students hands-on programming skills and best practices related to Data Science and Artificial Intelligence. It is a laboratory course in which students will develop programming skills in loading, cleansing, transforming, modeling and visualizing data.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Analyze the data using data analytic tools
2. Manipulate data sets programmatically
3. Perform exploratory data analysis programmatically
4. Apply basic and advance text processing techniques to unstructured data sets
5. Visualize data sets effectively
6. Perform basic statistical analyses programmatically
7. Build data-driven predictive models

Prerequisites: None

Course Outline:

I. Fundamentals
   1. Python programming
   2. The Python toolset
II. Working with data
   1. Numerical computation using numpy
   2. Data manipulation using pandas
   3. Exploratory data analysis
   4. Text processing with nltk
III. Data visualization
   1. Matplotlib
   2. Pandas
   3. Visdom
IV. Statistics
   1. Random variables
   2. Probability distributions
   3. Hypothesis testing using scipy and statsmodels
V. Machine learning tools
   1. Scikit-learn
   2. Pytorch

Laboratory Session(s): Each topic listed above is a series of lab sessions.

Learning Resources:

Textbooks: No specific textbook. Lab manuals and online resources will be used.

Reference Books:

Downey, A. (2014), *Think Stats*, 2nd edition, O'Reilly.

Geron, A. (2017), *Hands-On Machine learning with Scikit-Learn & TensorFlow*, O'Reilly.

McKinney, W. (2013), *Python for Data Analysis*, O'Reilly.

VanderPlas, J. (2016), *Python Data Science Handbook: Essential Tools for Working with Data*, O'Reilly.

Journals and Magazines:

IEEE Transactions on Knowledge and Data Engineering, IEEE
ACM/IMS Transactions on Data Science, ACM
Journal of Machine Learning Research (JMLR), Microtome
Neural Networks, Elsevier

Others:

Python tutorials available online: https://docs.python.org/3/tutorial/
Jupyter notebook tutorials available online: https://ipython.org/documentation.html
Numpy tutorials available online:https://numpy.org/doc/stable/
Pandas tutorials available online: https://pandas.pydata.org/docs/
Nltk tutorials available online: https://www.nltk.org
Matplotlib tutorials available online: https://matplotlib.org/contents.html
Visdom tutorials available online: https://github.com/facebookresearch/visdom
Scikit-learn tutorials available online: https://scikit-learn.org/stable/user_guide.html
Pytorch tutorials available online: https://pytorch.org/tutorials/

Teaching and Learning Methods:

1. Use of online tutorials: Students will make use of online tutorials for self-learning.

2. Laboratory sessions: Students will be required to perform a series of exercises and submit a lab report.
3. Homework: Several homework exercises requiring students to apply the knowledge acquired from lab and discussion will be assigned and graded.

Time Distribution and Study Load:

- Laboratory sessions: 90 hours.
- Self study and laboratory report preparation: 45 hours.

Evaluation Scheme:

1. Attendance and laboratory procedure completion: 20%
2. Laboratory reports: 60%
3. Final practical examination 20%

A grade of "A" indicates successful completion of all procedures and excellent and insightful understanding of the techniques introduced in the laboratory; "B" indicates mostly successful completion and a good understanding of the techniques; "C" indicates barely acceptable completion and understanding; and "D" indicates inability to complete many procedures and/or poor understanding of the techniques.

Instructor(s): Dr. Chaklam Silpasuwanchai

AT82.02 Data Modeling and Management, 3(2-3) Semester: August

Objective: The course emphasizes on emerging data models and technologies suitable for managing different types and characteristics of data. Students will develop skills in analyzing, evaluating, modeling and developing database applications with concerns on both technical and business requirements.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Explain data modeling and management concepts.
2. Design and organize various types of data using a relational and non-relational data models.
3. Analyze the characteristics and requirements of data and select an appropriate data model.
4. Evaluate, implement and perform frequent data operations (CRUD: create, read, update and delete) on relational and NoSQL databases.
5. Describe the concepts and the importance of big data, data security, privacy and governance.
6. Interpret the concepts and the importance of data engineering and data visualization.

Prerequisites: None

Course Outline:

I. Recall: Relational Data Model and Management
  1. Relational Model Concepts
  2. SQL
  3. Relational Database Design and Normalization
  4. Relational Database Management Systems (RDBMSs)

II. NoSQL Data Modeling and Management
  1. NoSQL Concepts and Characteristics
  2. Major Categories of NoSQL Data Models
  3. NoSQL Database Design
  4. NoSQL Features and Operations

III. Data Distribution
  1. Data Sharding and Replication Models
  2. CAP Theorem

IV. Transaction Processing and Consistency Models
  1. Transaction Processing Concepts
  2. ACID Model
  3. BASE Model

V. Large Scale Data Handling
  1. Big Data characteristics
  2. Big Data Modeling and Management

VI. Data Engineering
  1. Business Understanding
  2. Data Acquisition and Understanding
  3. Data Cleansing
  4. Data Preparation, Transformation and Feature Engineering

VII. Introduction to Related Topics
  1. Data Security
  2. Data Privacy and Legal Issues,
  3. Data Governance: Social and Ethical Issues, Biasness (gender, religions, etc.)

Laboratory Session(s):

1. Relational database model and tools
2. NoSQL data stores and tools, CRUD operations
   a. Key-value model
   b. Column-family model
   c. Document model
   d. Graph model
3. API development/usage
4. Data engineering

Learning Resources:

Textbooks:

A. Meier and M. Kaufmann: SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management, Springer, 2019, ISBN 978-3658245481

Reference Books:

M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, O'Reilly, 2017, ISBN 978-1449373320

D. Sullivan, NoSQL for Mere Mortals, Addison-Wesley, 2015, ISBN 978-0-1340-2321-2

P. Sadalage and M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2013, ISBN 978-0-3218-2662-6

E. Redmond and J. R. Wilson, Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement, 2012, ISBN 978-1-93435-692-0

G. Harrison,Next Generation Databases: NoSQL and Big Data, Apress, 2015, ISBN 978-1-4842-1329-2

I. Robinson, J. Webber and E. Eifrem, Graph Databases: New Opportunities for Connected Data, 2/E, O'Reilly, 2015, ISBN 978-1-491-93200-1

R. Elmasri and S. Navathe: Fundamentals of Database Systems, 7/E, Addison-Wesley, 2015

Journals and Magazines:

IEEE Transactions on Knowledge and Data Engineering, IEEE

ACM Transactions on Database Systems, ACM

ACM Transactions on Information Systems, ACM

Teaching and Learning Methods:

1. Lectures
2. Discussion and case studies
3. Laboratory sessions: Students will be required to perform a series of exercises in data analysis and submit a lab report.
4. Homework: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
5. Project: Students will propose and execute a plan for a significant data modeling project in groups. Students should execute their projects independently under the guidance of the instructor and make a formal presentation of the results.

Time Distribution and Study Load:

- Lectures: 30 hours.
- Laboratory sessions: 45 hours.
- Self study: 30 hours.
- Homework: 30 hours.
- Project work: 30 hours.

Evaluation Scheme:

1. Midterm exam: 25%
2. Assignments and laboratory work: 20%
3. Project: 30%
4. Final examination: 25%

A grade of "A" indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; "B" indicates a good understanding of the key concepts and ability to implement basic techniques; "C" indicates barely acceptable understanding and implementation ability; and "D" indicates poor understanding and implementation ability.

Instructor(s): Dr. Chutiporn Anutariya

AT82.03 Machine Learning, 3(2-3) Semester: August

Objective: The course provides students from a variety of science, engineering, and management backgrounds with a strong foundation in the fundamentals of machine learning and prepares them to perform R&D involving machine learning techniques and applications.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Formulate a practical data analysis and prediction problem as a machine learning problem.
2. Plan for data acquisition considering the characteristics of the data set required for a particular machine learning problem.
3. Train and test supervised regression and classification models, unsupervised learning and density estimation models, and reinforcement learning models.
4. Integrate a trained machine learning model into an online software system.

Prerequisites: None

Course Outline:

I. Introduction to Machine Learning

II. Supervised Learning
   1. Linear regression, logistic regression, and generalized linear models
   2. Generative probabilistic models
   3. Convex optimization and quadratic programming
   4. Support vector machines
   5. Decision trees and ensemble models
   6. Non-parametric methods

III. Neural Networks
   1. Perceptrons and inspiration from neuroscience
   2. Multilayer neural networks and backpropagation
   3. Optimization techniques, best practices, loss curve analysis

IV. Learning Theory
   1. Bias-variance tradeoff
   2. Regularization, model selection, and feature selection
   3. Generalization bounds and VC dimension

V. Unsupervised Learning
   1. Clustering: k-means, Gaussian mixture models
   2. Principal components analysis
   3. Independent components analysis
   4. Autoencoders

VI. Reinforcement Learning
   1. Markov decision processes and the Bellman equations
   2. Value iteration, policy iteration
   3. Q-learning

Laboratory Session(s):

1. Linear regression models
2. Logistic regression
3. Support vector classification
4. Decision trees
5. Single-layer and multi-layer neural networks
6. Multi-layer back-propagation, regularization, hyperparameter search
7. Model selection, feature selection
8. Clustering with k-means and GMMs
9. Principal components analysis and autoencoders
10. Value iteration and policy iteration
11. Q-learning
12. Deploying a machine learning model

Learning Resources:

Textbooks: No designated textbook, but class notes and handouts will be provided.

Reference Books:

Mitchell, T. (1997), *Machine Learning*, McGraw-Hill.

Bishop, C. (2006), *Pattern Recognition and Machine Learning*, Springer.

Goodfellow, I., Bengio, Y., and Courville, A. (2016), *Deep Learning*, MIT Press.

Hastie, T., Tibshirani, R., and Friedman, J. (2016), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition, Springer.

Sutton, R.S. and Barto, A.G. (2018), *Reinforcement Learning: An Introduction*, 2nd edition, MIT Press.

Journals and Magazines:

IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)

Journal of Machine Learning Research (JMLR). Microtome

Pattern Recognition. Elsevier

Neural Networks. Elsevier

IEEE Transactions on Knowledge and Data Engineering

<u>Others</u>:

Proceedings of the *Advances in Neural Information Processing Systems (NeurIPS)* conference. Neural Information Systems Foundation, Inc.

Proceedings of the *International Conference on Machine Learning (ICML)*. International Machine Learning Society.

Lecture notes.

<u>Teaching and Learning Methods</u>:

1. Use of online resources outside of class: Students will be periodically assigned online video lectures prior to the face-to-face lecture.
2. Lectures
3. In-class tutorials: Tutorials on important data analysis and modeling tools will be given in class periodically.
4. Laboratory sessions: Students will be required to perform a series of exercises in data analysis and submit a lab report.
5. Homework: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
6. Project: Students will propose and execute a plan for a significant machine learning project in groups of 1-3. Students should formulate their data analysis problems independently under the guidance of the instructor, deploy a prototype, and make a formal present the results.

<u>Time Distribution and Study Load</u>:

- Lectures: 30 hours.
- Laboratory sessions: 45 hours.
- Tutorials: 15 hours.
- Self study: 65 hours.
- Homework: 35 hours.
- Project work: 35 hours.

<u>Evaluation Scheme</u>:

1. Term project: 25%
2. Homework and laboratory reports: 25%
3. Midterm examination: 20%
4. Final examination: 30%

A grade of “A” indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; “B” indicates a good understanding of the key concepts and ability to implement basic techniques; “C” indicates barely acceptable understanding and implementation ability; and “D” indicates poor understanding and implementation ability.

Instructor(s): Prof. Matthew Dailey

AT82.04 Business Intelligence and Analytics, 3(3-0) Semester: January

Objective: This course will give students an understanding of the principles and practices of business intelligence and data analytics to support organizations in conducting their business in a competitive environment, in order to support better business decision making and capture new business opportunities.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Explain the concepts characteristics of BI and data analytics
2. Describe multiple business problem/decision making domains requiring BI and data analytics
3. Apply BI and data analytic tools and technologies to develop BI applications
2. Integrate BI applications with other information systems as part of a business process5. Manage a BI project for an organization
3. Interpret big data analytics and applications

Prerequisite: Data Modeling and Management

Course Outline:

I. Introduction to Business Intelligence
   1. BI Definition
   2. BI Concepts
   3. Business Intelligence, Analytics, and Data Science
   4. Business Intelligence to Support Decisions

II. Data Warehousing for BI
   1. DW design
   2. Multidimensional data modelling and analysis
   3. ETL process

III. Categories of Data analytics:
   1. Descriptive Analytics
   2. Predictive Analytics
   3. Prescriptive Analytics

IV. Descriptive Analytics
   1. Descriptive Statistics
   2. Business Performance Management
   4. Data Visualization and Dashboard Design

V. Predictive Analytics
   1. Data Mining (Text Analytics and Text Mining, Web Analytics, Web Mining, and Social Analytics)
   2. Predictive Modeling

VI. Overview of Prescriptive Analytics
   1. Optimization
   2. Multi-Criteria Systems

VII. Technical Aspects
   1. BI Architecture
   2. BI Tools and Technologies

VIII. BI Applications
1. BI Maturity
2. BI Strategies
3. BI Project (case study)

IX. Overview of Big Data
1. Big Data Analytics
2. Example of Big Data Applications

Laboratory Session(s): None

Learning Resources:

Text Book:

Business intelligence, analytics, and data science, by Ramesh Sharda;Dursun Delen; Efraim Turban, Pearson Publisher, 2018

Recommended books:

1. Business Analytics (2nd Ed.) by James Evans, Pearson, 2017.
2. Business Analysis for Business Intelligence (1st Ed) by Bert Brijs, Auerbach Publications, 2013.
3. Business Intelligence Guidebook (1st Ed) by Rick Sherman, Morgan Kaufmann, 2014.
4. Fundamentals of Business Intelligence by Wilfried Grossmann and Stefanie Rinderle-Ma, Springer, 2015.

Journals:

Decision Support Systems, Elsevier.

International Journal of Business Intelligence and Data Mining, Inderscience.

International Journal of Business Intelligence Research, IGI Global.

Journal of Big Data, Springer.

Teaching and Learning Methods:

1. Lectures
2. Real-world case studies
3. Homework: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
4. Project: Students will propose and execute a plan for a significant business intelligence project in groups. Students should execute their projects independently under the guidance of the instructor and make a formal presentation of the results.

5. Self learning

Time Distribution and Study Load:

- Lectures: 45 hours.
- Self study: 75 hours.
- Homework: 30 hours.
- Project work: 30 hours.

Evaluation Components

1. Mid-term exam 20%
2. Final exam 20%
3. Homework assignments 20%
4. Course Project 40%

A grade of “A” indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; “B” indicates a good understanding of the key concepts and ability to implement basic techniques; “C” indicates barely acceptable understanding and implementation ability; and “D” indicates poor understanding and implementation ability.

Instructor(s): Dr. Vatcharaporn Esichaikul

AT82.06 Artificial Intelligence: Semester: August

Problem Solving and Planning, 3(2-3)

Objective: The course aims to introducing the students for the fundamentals of Artificial Intelligence and its techniques. Students will be exposed to several techniques on planning and decision procedures ranging from precise to uncertain and temporal reasoning with applications to intelligent agents.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Demonstrate fundamental insights into practical planning and decisionmaking procedures
2. Apply algorithms for reasoning under uncertainty
3. Apply planning techniques into intelligent agents

Prerequisites: None

Course Outline:

I. Introduction to AI
   1. Definition of Artificial intelligence
   2. History of artificial intelligence

II. Intelligent agents
   1. Concepts of Rationality
   2. Structure of Agents

III. Planning and decision making
   1. Decision trees and search techniques
   2. Heuristic algorithms

IV. Constrained planning
   1. Recall of propositional and predicate logic
   2. Unification and resolution
   3. Prolog and constraint solvers

V. Planning under uncertainty
   1. Bayesian networks
   2. (Partially observable) Markov decision networks

VI. Temporal planning
   1. Temporal reasoning
   2. Scheduling

Laboratory Session(s):

Lab sessions for assignments on II, IV, V, and VI

Lab sessions to explain and train the students hand-on understanding and skill with well-known planners:

- FF plan system (https://fai.cs.uni-saarland.de/hoffmann/ff.html) for planning with heuristic algorithms
- Constraint and Prolog solvers for Constraint Solvers (https://www.gecode.org, https://www.swi-prolog.org)
- Planner with Uncertainty (https://www.cs.nmsu.edu/~tson/systems.html)
- Planners with temporal reasoning (http://www.cs.toronto.edu/tlplan/tlplan.shtml, https://www.aaai.org/ojs/index.php/aimagazine/article/view/1581).

Learning Resources:

Textbooks: No designated textbook, but lecture notes will be provided.

Reference Books:

M. Valatti and D. Kitchin, Knowledge Engineering Tools and Techniques and AI Planning, Springer verlag, 2020

P. Haslum and N. Lipovetzky, An introduction to the Planning Domain definition Language, Morgan & Claypool, 2019

M. Gallab, D. Nau and P. Traverson, Automated Planning and Acting, Cambridge University Press, 2016

Russel, S. and Norvig, P. (2013), *Artificial Intelligence: A Modern Approach*, 3rd edition, Pearson.

Ghallab, M., Nau, D., and Traverso, P. (2004) *Automated Planning: Theory & Practice*, 1st edition,

Morgan Kaufmann Publishers and Elsevier.

Bratko, I. (2011), *Prolog Programming for Artificial Intellige*nce, 4th edition, Pearson.

Teaching and Learning Methods:

1. Lectures
2. Self learning: for Prolog
3. Laboratory sessions: Students will be required to perform a series of procedures in the laboratory and submit a lab report.
4. Homework: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
5. Project: Students will propose and execute a plan for a significant project in groups. Students should execute their projects independently under the guidance of the instructor and make a formal presentation of the results.

Time Distribution and Study Load:

- Lectures: 30 hours.
- Laboratory sessions: 45 hours
- Self study: 45 hours.
- Homework: 30 hours.
- Project work: 30 hours.

<u>Evaluation Components</u>

1. Midterm exam (20%)
2. Final exam (30%)
3. Homework (20%)
4. Project (30%)

A grade of "A" indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; "B" indicates a good understanding of the key concepts and ability to implement basic techniques; "C" indicates barely acceptable understanding and implementation ability; and "D" indicates poor understanding and implementation ability.

<u>Instructor(s):</u> Prof. Phan Minh Dung

AT82.05 Artificial Intelligence: Semester: Inter-Sem

Natural Language Understanding, 3(3-0)

Objective: Introducing students to the linguistic knowledge of natural languages together with the algorithms and technologies for processing them. Key linguistic concepts of words, morphology, parts-of-speech, syntax and semantics are presented together with algorithms and technologies like regular expressions, finite automata, context-free grammars, unification, first-order logic, lambda-notations, hidden Markov models as well as other rule-based or statistical algorithms.

Learning Outcomes: Students, on completion of the course, would be able to

1. Apply knowledge on regular expression, part-of-speech to develop algorithms for spell checkers in text editors.
2. Apply knowledge on regular expression, part-of-speech and parsing to develop parsers for different natural language applications.
3. Develop algorithms for reasoning in natural language by translating the respective domain language into formal language like first-order logics.
4. Develop algorithms for information extractions from natural language texts.
5. Develop algorithms for Querying-Answering and Dialogue systems in natural language.

Prerequisite: None

Course Outline:

I Introduction to the ambiguity of natural language.

II Words

1. Morphology and Parts of Speech
2. Algorithms: Regular Expression, Finite Automata, Hidden Markov Models
3. Language Models: N-Grams

III Syntax

1. Context-Free Grammars
2. Formal Grammars of English
3. Syntactic Parsin
4. Statistical Parsing
5. Features and Unification

IV Semantics

1. Lexical Semantics
2. Sentence Semantics
3. Discourse

Laboratory Session(s): None

Learning Resources:

Text Book: No designated textbooks, but lecture notes will be provided.

References:

E.M. Bender and A. Lascarides, Linguistic Fundamental for Natural Language Processing: 100 Essentials from Semantics and Pragmatics, Morgan & Claypool Publishers, 2019

S. Vajjala and B. Majumder, Practical natural Language Processing, O'Reilly, 2020

D. Jurafsky, J. M. Martin, Speech and Language Processing, Second Edition, Pearson International, 2008

P. M. Nugues, Language Processing with Perl and Prolog, Second Edition, Springer Verlag, 2014

A. Carnie, Syntax: A Generative Introduction, Third Edition, Wiley-Blackwell, 2013

M. Steedman, The Syntactic Process, MIT press, 2001

M. Steedman, Surface Structure and Interpretation, MIT press, 1996

M. Bramer, Logic Programming with Prolog, Second edition, Springer Verlag, 2013

P. Blackburn, J. Bos, Representation and Inference for Natural Language, CSLI Publication, 2008

Journals and Magazines:

The Journal of Artificial Intelligence, Elsevier Science

Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), IJCAI Organization

Proceedings of Conferences of Associations for Advancement of Artificial Intelligence (AAAI), AAAI Organization

Teaching and Learning Methods:

1. Lectures
2. Homework: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
3. Project: Students will propose and execute a plan for a significant natural language understanding project in groups. Students should execute their projects independently under the guidance of the instructor and make a formal presentation of the results.

Time Distribution and Study Load:

- Lectures: 45 hours.
- Self study: 100 hours.
- Homework: 10 hours.
- Project work: 25 hours.

Evaluation Scheme: The final grade will be from the following constituent parts:

1. Midterm exam (20%)
2. Final exam (50%)
3. Assignments and project (30%)

Exams are open book.

A grade of "A" indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; "B" indicates a good understanding of the key concepts and ability to implement basic techniques; "C" indicates barely acceptable understanding and implementation ability; and "D" indicates poor understanding and implementation ability.

Instructor(s): Prof. Phan Minh Dung

AT82.08 Computer Vision, 3(2-3) Semester: Inter-Sem

Course Objective: The course provides students with research and development skills in the image processing, geometry, and statistical inference tools necessary for extracting useful information about the world from two-dimensional images, including applications to robot vision, intelligent video surveillance and monitoring, optical character recognition, and human-computer interfaces.

Learning Outcomes: The students on the completion of this course would be able to:

1. Develop computer programs to find point correspondences between different images of a 3D scene;
2. Use noisy point correspondences between images to estimate projective transformations between planes, camera positions and orientations, and the 3D structure of a scene;

3. Apply machine learning techniques for classification to problems involving segmentation, detection, and recognition of people and other objects in video sequences as well as optical character recognition;
4. Apply sequential state estimation techniques to problems involving tracking of people and other objects in video sequences;
5. Implement and evaluate state-of-the-art machine vision algorithms described in the primary academic literature;
6. Integrate the necessary statistical estimation, image processing, and machine learning tools with a custom-designed algorithm to provide a complete solution to an image or video processing problem.

Prerequisites: None.

Course Outline:

I. Introduction
   1. The goal: Giving machines the gift or sight
   2. Computer vision systems around us today
   3. Overview of computer vision research today

II. Projective geometry
   1. 2D projective geometry
   2. 3D projective geometry
   3. Rigid transformations

III. Statistical estimation
   1. Linear methods
   2. Singular Value Decomposition (SVD)
   3. Nonlinear methods
   4. Robust methods

IV. Cameras
   1. Finite cameras
   2. General cameras
   3. Camera parameter estimation (calibration)

V. Two-view stereo
   1. Epipolar geometry and the fundamental matrix
   2. Computing the fundamental matrix
   3. Interest points for sparse correspondence estimation
   4. Stereo rectification
   5. 3D structure computation

VI. N-view reconstruction
   1. Bundle adjustment

2. Factorization
3. Metric upgrade and autocalibration

VII. Machine learning

1. The Bayesian approach and empirical risk minimization
2. Feature selection and classical computer vision methods
3. CNNs and deep learning methods for computer vision

VIII. Sequential state estimation

1. Kalman filters and extended Kalman filters
2. Deep learning approaches to single and multiple object tracking

Laboratory Session(s):

1. OpenCV and Octave
2. Estimating homographies
3. Estimating cameras (calibration)
4. Structure from motion
5. CNNs
6. Learning object classification models
7. Learning semantic segmentation models
8. Learning instance segmentation models
9. Single object tracking
10. Multiple object tracking

Learning Resources:

Textbooks:

R. Hartley and A. Zisserman (2004), Multiple View Geometry in Computer Vision, 2nd edition, Cambridge University Press.

I. Goodfellow, Y. Bengio, and A. Courville (2016), Deep Learning, MIT Press, electronic copy available from Amazon.com.

References:

R. Szeliski (2010), Computer Vision: Algorithms and Applications, Springer, available on Web.

C. Bishop (2006), Pattern Recognition and Machine Learning, Springer.

A. Kaehler and G. Bradski (2016), Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, O'Reilly.

Journals and Magazines:

- International Journal of Computer Vision (IJCV), Springer.
- IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)
- Image and Vision Computing, Elsevier.
- Pattern Recognition, Elsevier.
- Pattern Recognition Letters, Elsevier.
- Computer Vision and Image Understanding, Elsevier.

Others:

High impact conference proceedings:

- IEEE Computer Vision and Pattern Recognition Conference (CVPR).
- International Conference on Computer Vision (ICCV). Springer.
- European Conference on Computer Vision (ECCV). Springer.
- Asian Conference on Computer Vision (ACCV). Springer.
- IEEE International Conference on Robotics and Automation (ICRA).
- IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Teaching and Learning Methods:

1. Use of online resources outside of class: Students will be periodically assigned online video lectures for self-study prior to the face-to-face lecture.
2. Lectures and in-class discussion
3. In-class exercises: The theoretical presentation of most topics will be followed by practical in-class exercises with discussion of the implementation techniques involved.
4. Paper presentations: Students will individually present research papers in class from the primary academic literature representing recent advances in the state of the art.
5. Laboratory reports: Students submit a laboratory report for each laboratory session. The report describes how the in-lab exercises and take-home assignments were solved and how the knowledge acquired from lecture and discussion was applied.
6. Project: Students propose and execute a plan for a significant computer vision project in groups of 1-3. Students should formulate their data analysis problems independently under the guidance of the instructor, deploy a prototype, and make a formal presentation of the results.

Time Distribution and Study Load:

- Lectures: 30 hours.
- Laboratory sessions: 45 hours
- Lab reports: 15 hours
- Self study: 60 hours.
- Project work: 30 hours.

Evaluation Scheme:

Homework:	20%
Presentations:	20%
Midterm:	20%
Final exam:	20%
Project:	20%

A grade of “A” indicates excellent and insightful understanding of the theoretical foundations of computer vision and the practical skills necessary to implement intelligent systems utilizing cameras; “B” indicates a good understanding of the theoretical foundations and the practical skills necessary to implement basic systems incorporating cameras; “C” indicates barely acceptable understanding of the foundations and some implementation ability; and “D” indicates poor understanding and implementation ability.

Instructor(s): Prof. Matthew N. Dailey and Dr. Mongkol Ekpanyapong

AT82.09 Human-Computer Interaction and Information Visualization, 3(2-3) Semester: January

Course Objective:

The objective of this course is to provide the concepts of HCI and user interfaces, focusing on user interface design and technologies. The students will learn principles and skills for designing interactive systems and Web-based applications.

Learning Outcomes: The students on the completion of this course would be able to:

- Explain the concepts of Human-Computer Interaction (HCI), user interface, user interface design,
- Apply interface techniques and technologies; Graphical User Interface (GUI), direct manipulation, menu
- Analyze the outcomes of interface design

Prerequisite: None

Course Outline:

I. Introduction to Human-Computer Interaction (HCI)

1. Human: Human Memory, Thinking, Individual Differences
2. Computer: Entry Devices, Positioning and Pointing Devices, Output Devices
3. Interaction: Models of Interaction, Ergonomics, Interaction Styles

II. Theories and Principles

1. High-level Theories
2. Object-Action Interface Model
3. Golden Rules of Interface Design

III. User Interface

1. Interface Widgets
2. Interactive Devices
3. Printed and Online Facilities

IV. User Interface Design

1. Design Development Process
2. Software Tools
3. User and Task Analysis

4. Multimodal Interfaces
5. Response Time and Display Rate
6. Presentation Style

V. Interface Techniques and Technologies

1. Graphical User Interface (GUI)
2. Direct Manipulation
3. Menu Selection
4. Form Fillin and Dialog Boxes
5. Command and Natural Languages
6. Multiple Windows
7. Hypermedia and World Wide Web
8. Virtual Environments

VI. Evaluation of Interface Design

1. Expert Review
2. Usability Testing
3. Acceptance Tests
4. Experiments

VII. Ubiquitous Computing Interaction

1. Interface Design for Handheld Devices
2. Handheld Usability

Laboratory Session(s): None

Textbook(s):

S. Mackenzie (2013), Human-Computer Interaction: An Empirical Research Perspective, Morgan Kaufmann (1st edition).

B. Shneiderman (2016), Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison Wesley (6th edition).

Reference Books:

J. Preece, H. Sharp, and Y. Rogers (2019), Beyond Interaction Design, John Wiley & Sons (5th edition)

D. Norman (2013), The Design of Everyday Things, Basic Book (2nd edition).

S. Krug (2014), Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, New Riders (3rd edition).

D. Kahneman (2013), Thinking, Fast and slow, Farrar, Straus and Giroux (1st edition).

Journals and Magazines:

- ACM Transactions on Computer Human Interaction, ACM
- CHI Conference Proceedings, ACM
- Communication of the ACM, ACM
- Human-Computer Interaction, Taylor and Francis
- IEEE Computer, IEEE
- International Journal of Human Computer Studies, Elsevier

Teaching and Learning Methods:

1. **Lectures and in-class discussion**
2. **Homework**: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
3. **Group project**: Students will propose and execute a plan for a significant project in groups. Students should formulate and execute their projects independently under the guidance of the instructor, and make a formal presentation of the results.

Time Distribution and Study Load:

- Lectures: 45 hours.
- Self study: 70 hours.
- Homework: 30 hours.
- Project work: 35 hours.

Evaluation Scheme:

The final grade will be computed from the following constituent parts:

Mid-semester exam (30%)

Final exam (30%)

Assignments/projects (40%)

Closed-book examination is used for both mid-semester and final exam.

A grade of “A” indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; “B” indicates a good understanding of the key concepts and ability to implement basic techniques; “C” indicates barely acceptable understanding and implementation ability; and “D” indicates poor understanding and implementation ability.

Instructor(s): Dr. Chaklam Silpasuwanchai

AT82.07 Artificial Intelligence: Knowledge Representation and Reasoning 3(3-0) Semester: January

Objective:

The course introduces students to the theories and methodologies of knowledge representation and reasoning in AI. Both the model and proof theoretic semantics of first-order logics (FOL) are presented. Examples and reasons for the weakness of FOL for practical reasoning are explained. Bayesian reasoning as an important tool in modelling uncertain reasoning is studied. Temporal reasoning, planning together with causal reasoning are presented. A new novel insight of practical reasoning as arguing is finally introduced together with theories of argumentation as a platform for modelling the arguing process. At last, application of practical reasoning in knowledge engineering is presented.

Learning Outcomes: Students, on completion of the course, would be able to:

- Apply the proof procedures in argumentation to build dialogue systems.
- Apply the algorithms in practical reasoning system to model dispute resolution systems.
- Apply the algorithms in temporal planning and casual reasoning to build dynamic planning system in dynamic domains.

Prerequisite: None

Course Outline:

I Introduction

1. Examples of commonsense reasoning, legal reasoning, social discourse and dialogues.
2. Qualitative and quantitative uncertainties in practical reasoning.
3. Intelligent agents

II First-Order Logic (FOL)

1. Syntax and Semantics of FOL
2. Inference and Proof Procedures in FOL
3. FOL is not Good Enough for Practical Reasoning: The Monotony of FOL and Non-Monotony of Practical Reasoning

III Bayesian Reasoning

1. Bayesian Rule and Inference
2. Independence and Bayesian Networks

IV Temporal Reasoning and Planning

1. Time for Planning
2. Planning with Chronicles

V Causal Reasoning, Actual Causation

1. Causal Modeling
2. Tests of Actual Causations

VI Practical Reasoning as Arguing

1. Reasoning in FOL as Constructing Arguments: Natural Deduction
2. Examples of Arguing
2. Logic Programming

VII Argumentation

1. Introduction: Examples and Definitions
2. Abstract Argumentation: Arguments and Attacks
3. Structured Argumentation

VIII Knowledge Engineering

1. Ontology and Description Logic
2. Knowledge Acquisition: Building Relevant Knowledge Bases
3. Arguments and Proof Procedures for Query-Answering and Dialogues

Laboratory Session(s): None

Learning Resources:

Text Book: No designate textbook, but the lecture notes will be provided.

References:

Russell S., Norvig P.Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, 2010

Halpern J., Actual Causality, MIT Press, 2016

Halpern J., Reasoning about Uncertainty, MIT Press, 2003

Chriswell I., Hodges W., Mathematical Logic, Oxford Uni. Press, 2007

Pearl J., Causality: Models, Reasoning, and Inference, Cambridge University Press, 2009

Pearl J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inferences, Morgan Kaufmann Publishers, 1988

M. Bramer, Logic Programming with Prolog, Second edition, Springer Verlag, 2013

Clocksin W.F., Mellish C.S, Programming in Prolog, Fifth Edition, Springer Verlag, 2003

Others:

- Instructors collection of Papers/Articles on Argumentation in Journals on Artificial Intelligence.

Teaching and Learning Methods:

1. **Lectures and in-class discussion**
2. **Homework**: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
3. **Group project**: Students will propose and execute a plan for a significant project in groups. Students should formulate and execute their projects independently under the guidance of the instructor, and make a formal presentation of the results.

Time Distribution and Study Load:

- Lectures: 45 hours.
- Self study: 100 hours.
- Homework: 20 hours.
- Project work: 15 hours.

Evaluation Scheme:

The final grade will be from the following constituent parts:

Mid-semester exam 20%

Final exam 55%

Assignments and projects 25%

Exams are open book.

A grade of "A" indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; "B" indicates a good understanding of the key concepts and ability to implement basic techniques; "C" indicates barely acceptable understanding and implementation ability; and "D" indicates poor understanding and implementation ability.

Instructor(s): Prof. Phan Minh Dung

AT82.10 Recent Trends in Machine Learning 3(2-3) Semester: January

Objective: The course builds on the content of Machine Learning, providing students with a deeper understanding of machine learning techniques and a wider variety of extant learning models. Students will be prepared to develop advanced machine learning applications and perform research at a state-of-the-art level.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Design, train, test, and deploy modern convolutional neural networks (CNNs).
2. Utilize the principles of adversarial learning to increase the robustness of a machine learning model.
3. Design, train, test, and deploy generative adversarial networks (GANs).
4. Utilize recurrent neural networks (RNNs) to model and predict time series.
5. Utilize deep neural networks to solve difficult tabula rasa reinforcement learning problems.
6. Apply state-of-the-art machine learning methods to solve problems in speech processing, speech synthesis, natural language understanding, natural language synthesis, computer vision, and intelligent agent design.

Prerequisites: None

Course Outline:

I. Overview of modern machine learning methods

II. Convolutional neural networks
   1. Fundamentals
   2. Inception modules
   3. Residual layers
   4. Squeeze and excitation
   5. Detection models
   6. Semantic segmentation models
   7. Instance-aware segmentation models

III. Transfer learning
   1. Inductive transfer learning
   2. Transductive transfer learning
   3. Unsupervised transfer learning

IV. Automatic learning
   1. Automated feature engineering
   2. Automated model selection
   3. Automated optimization algorithm selection

V. Deep unsupervised learning
   1. Generative adversarial networks (GANs)
   2. Cycle GANs
   3. Wasserstein GANs
   4. Variational autoencoders

VI. Practical techniques for deep learning models

1. Weight initialization
2. Dropout
3. Adam optimization
4. Batch normalization

VII. Time series processing

1. Hidden Markov models (HMMs)
2. Recurrent neural networks (RNNs) and backpropagation through time
3. Word embedding for natural language processing
4. Long short term memory (LSTM) units
5. Gated recurrent units (GRUs)
6. Attention mechanisms for RNNs

VIII. Deep Reinforcement learning

1. Policy gradients
2. Actor/critic methods
3. Imitation learning
4. Exploration/exploitation
5. Meta learning
6. Monte Carlo methods

IX. Applications

1. Speech recognition
2. Speech synthesis
3. Conversational agents
4. Recommendation systems
5. Anomaly detection
6. Computer vision systems

Laboratory Session(s):

1. Preparing the environment for machine learning tools
2. CNNs and residual layers
3. Generative adversarial networks (GANs)
4. Deep learning techniques
5. Introductory time series processing
6. Time series processing with LSTMs and GRUs
7. Deep reinforcement learning
8. Deep speech recognition
9. Recommendation systems
10. Anomaly detection
11. Computer vision

Learning Resources:

Textbooks: No designated textbook. Emphasis is on recent papers in major machine learning conferences. Class notes and handouts will be provided.

Reference Books:

Goodfellow, I., Bengio, Y., and Courville, A. (2016), *Deep Learning*, MIT Press.

Sutton, R.S. and Barto, A.G. (2018), *Reinforcement Learning: An Introduction*, 2nd edition, MIT Press.

Journals and Magazines:

IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). IEEE

Journal of Machine Learning Research (JMLR). Microtome

Others:

Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). IEEE

Proceedings of the *Advances in Neural Information Processing Systems (NeurIPS)* conference. Neural Information Systems Foundation, Inc.

Proceedings of the *International Conference on Machine Learning (ICML)*. International Machine Learning Society.

Lecture notes, posted online.

Teaching and Learning Methods:

1. **Use of online resources outside of class:** Students will be periodically assigned online video lectures prior to the face-to-face lecture.
2. **Lectures**
3. **In-class tutorials**: Tutorials on important data analysis and modeling tools will be given in class periodically.
4. **Laboratory sessions**: Students will be required to perform a series of exercises in data analysis and submit a lab report.
5. **Homework**: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
6. **Project**: Students will propose and execute a plan for a significant machine learning project in groups of 1-3.

Time Distribution and Study Load:

- Lectures: 30 hours.
- Laboratory sessions: 45 hours.
- Self study: 45 hours.
- Homework: 30 hours.
- Project work: 30 hours.

Evaluation Components

1. Term project (20%)
2. Midterm examination (20%)
3. Final examination (20%)
4. Homework (20%)
5. Lab reports (20%)

A grade of "A" indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; "B" indicates a good understanding of the key concepts and ability to implement basic techniques; "C" indicates barely acceptable understanding and implementation ability; and "D" indicates poor understanding and implementation ability.

Instructor(s): Prof. Matthew N. Dailey and Dr. Mongkol Ekpanyapong

AT82.11 Multicriteria Optimization and Decision Analysis 3(3-0) Semester: January

Objective: Multicriteria optimization and decision analysis deals with various aspects of finding optimal solutions in problems with multiple decision alternatives and conflicting objectives. This course will provide students an understanding of the decision making process and multicriteria decision analysis methods and optimization processes.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Describe the decision making processes typically used by organizations.
2. Formulate a decision making scenario and apply multicriteria decision analysis methods to decision making problems.
3. Evaluate and formulate different types of mathematical programming problems including formulations with constraints and multiple objectives.
4. Apply methods of multicriteria optimization and decision analysis in practice to real world problem domains.

Prerequisites: None.

Course Outline:

I. Decision making
   1. Decision making processes in organizations
   2. The problem of multiple criteria

II. Multicriteria decision analysis methods and approaches
   1. Analytic hierarchy process (AHP)
   2. Analytic network process (ANP)
   3. Multiattribute utility theory

III. Multicriteria optimization
   1. Multi-objective optimization problems
   2. Pareto optimal solutions
   3. Scalarization methods
   4. Multicriteria linear programming

5. Multicriteria integer programming

IV. Applications (examples)
   1. Health care
   2. Project management
   3. Water management

V. Tools and software
   1. Expert Choice
   2. Decisionarium

Laboratory Session(s): None

Learning Resources:

Textbooks:

Ishizaka, A., and Nemery, P. (2013). *Multi-Criteria Decision Analysis: Methods and Software,* John Wiley & Sons.

Reference Books:

Ehrgott, M. (2000), *Multicriteria Optimization*, Springer.

Kaliszewski, I., Miroforidis, J., and Podkopaev, D. (2016), *Multiple Criteria Decision Making by Multiobjective Optimization: A Toolbox*, Springer.

Journals and Magazines:

Journal of Multi-Criteria Decision Making, Wiley.

Decision Support Systems, Elsevier.

Journal of behavioral Decision Making, Wiley.

International Journal of Management and Decision Making, Inderscience.

Others:

Lecture notes provided online.

Teaching and Learning Methods:

1. **Lectures**
2. **Self learning:** Students will make use of online materials for self learning.
3. **Real-world case studies:** Case studies that require students to apply the knowledge acquired in class will be assigned and graded.
4. **Homework**: Several homework exercises requiring students to apply the knowledge acquired from lecture and discussion will be assigned and graded.
5. **Project**: Students will propose and execute a plan for a significant project in groups.

Time Distribution and Study Load:

- Lectures: 45 hours.
- Self study: 70 hours.
- Homework: 30 hours.
- Project work: 35 hours.

Evaluation Components

1. Mid-term exam 20%
2. Final exam 20%
3. Assignment 20%
4. Course Project 40%

A grade of "A" indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; "B" indicates a good understanding of the key concepts and ability to implement basic techniques; "C" indicates barely acceptable understanding and implementation ability; and "D" indicates poor understanding and implementation ability.

Instructor(s): Dr. Vatcharaporn Esichaikul

AT82.12 Software Development and Project Management 3(3-0) Semester: January

Objective: The course emphasizes modern and important software development, software processes, and project management. Student will tailor the software development process and project management for DS&AI projects, including planning, iterative development, test driven development, continuous integration/continuous delivery, versioning, and deliverables. The course provides examples and cases of how to apply the knowledge to the problems in DS&AI domains.

Learning Outcomes: Students, on successful completion of the course, will be able to

1. Explain the importance of software development and project management,
2. Explain how model-driven development works in a DevOps and agile environments,
3. Create model and data versioning,
4. Apply the principles of project management to DS & AI project.

Prerequisites: None

Course Outline:

I. Software Development and Software Process
   1. Introduction to Modern Software Process
   2. Agile practices and frameworks
   3. Model-driven development
   4. Test-Driven Development (TDD)
   5. Test Automation
   6. Continuous Integration/Continuous Delivery (CI/CD)
   7. Configuration Management
   8. DevOps

II. Project Management
   1. Project Integration Management
   2. Project Scope Management
   3. Project Time Management
   4. Project Cost Management
   5. Project Quality Management
   6. Project Human Resources Management
   7. Project Communications Management
   8. Project Risk Management
   9. Project Procurement Management
   10. Project Stakeholder Management

III. DS & AI Project Management
   1. Business Domain
   2. Healthcare Domain
   3. Smart City and Tourism Domain

Laboratory Session(s): None

Learning Resources:

Textbooks:

Project Management Institute. (2017*). A Guide to the Project Management Body of Knowledge (Pmbok Guide)*, 6th edition, The Stationery Office Ltd.

Reference Books:

Beck, K. and Andres, C. (2004). *Extreme Programming Explained: Embrace Change: Embracing Change*, 2nd Edition, Addison-Wesley Professional.

Forsgren, N., Humble, J., and Kim, G. (2018). *Accelerate: The Science of Lean Software and Devops: Building and Scaling High Performing Technology Organizations*, 1st Edition, IT Revolution Press.

Rubin, K.S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature): A Practical Guide To The Most Popular Agile Process (Addison-Wesley Signature Series (Cohn))*, 1st edition, Addison-Wesley Professional.

Humble, J. and Farley, D. (2010), *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler))*, 1st edition, Addison-Wesley Professional.

Journals and Magazines:

IEEE Transactions on Software Engineering (TSE), IEEE

Project Management Journal, SAGE

Journal of Systems and Software, Elsevier

Others:

Lecture notes, posted online.

Teaching and Learning Methods:

1. **Use of online resources outside of class:** Students will be periodically assigned online video lectures prior to the face-to-face lecture.
2. **Lectures**
3. **In-class tutorials**: Tutorials on important tools will be given in class.
4. **Individual Projects**: small individual projects that require students to apply the knowledge acquired from lecture and discussion will be assigned and graded
5. **Group Project**: Students will propose and execute a plan for a data science or artificial intelligence application project in groups of 3-4.

Time Distribution and Study Load:

- Lectures: 45 hours.
- Self study: 70 hours.
- Individual projects: 30 hours.
- Group project work: 35 hours.

Evaluation Components

1. Mid-sem Exam: 25%
2. Assignments: 20%
3. Project: 30%
4. Final Exam: 25%

A grade of "A" indicates excellent and insightful understanding of the key concepts and ability to implement sophisticated systems; "B" indicates a good understanding of the key concepts and ability to implement basic techniques; "C" indicates barely acceptable understanding and implementation ability; and "D" indicates poor understanding and implementation ability.

Instructor(s): Dr. Chutiporn Anutariya

For more information;

School of Engineering & Technology

https://set.ait.ac.th/

Computer Science

https://set.ait.ac.th/programs/information-and-communications-technologies/computer-science-cs/

Information Management

https://set.ait.ac.th/programs/information-and-communications-technologies/information-management-im/

Data Science and Artificial Intelligence

https://set.ait.ac.th/programs/information-and-communications-technologies/data-science-and-ai/

Contact person;

Prof. Matthew N. Dailey	email: mdailey@ait.ac.th
Prof. Phan Minh Dung	email: dungpm@ait.ac.th
Dr. Vatcharaporn Esichaikul	email: vatchara@ait.ac.th
Dr. Chutiporn Anutariya	email: chutiporn@ait.ac.th
Dr. Chaklam Silpasuwanchai	email: chaklam@ait.ac.th
Olivier Nicole	email: on@ait.ac.th
Siriporn Nanthasing	email: siripornn@ait.ac.th
Sireekant Thanwongpan	email: sireekant@ait.ac.th

Contact address;

Asian Institute of Technology

Computer Science & Information Management
58 Moo 9, Km.42, Paholyothin Highway,
Klong Luang, Pathumthani 12120 Thailand