VINUNIVERSITY

CURRICULUM FRAMEWORK

MASTER OF SCIENCE

Computer Science

PROGRAM CODE: 8480101

Applicable from Academic Year 2025-2026

(Released along with Decision No 589/2025/QĐ-VUNI dated 19/09/2025 by Provost of VinUniversity)

Records of changes

Version	Published date	Effective Date	Approved by	Description of changes
1	…	…	Developed by: Curriculum Development Committee Reviewed by: Curriculum Review Committee Endorsed by: Scientific and Educational Committee Approved by: Provost (Decision No. …./…./ QĐ-VUNI dated ….. by Provost of VinUniversity).	First Release

Table of Contents

Contents

1. PROGRAM OVERVIEW

1.1.Program Profile

Name of the degree	Master of Science in Computer Science
Name of the program	Master of Science in Computer Science
Orientation	Research
Program Code	8480101
Vietnam Qualifications Framework Level	7
Length of Program	2 years
Mode of Delivery	Full-time
Language of Delivery	English
Total credits	60 credits
Home College	College of Engineering and Computer Science

1.2.Program Purpose

The purpose of the programs is to develop computer scientists with an in-depth understanding of underlying computer science principles and significant exposure to practice and research activities, which will allow them to contribute to society as creative, innovative and personally well-rounded, proactively leading and advancing it. After graduation, graduates will be able to work in organizations that design cutting-edge technologies, research and development and in academia.

1.3. Program Educational Objectives and Program Learning Outcomes

1.3.1. Program Educational Objectives

The educational objectives of the Master of Science in Computer Science program are within a few years after graduation, graduates are expected to:

PEO1: Demonstrate advanced technical competence in computer science by applying cutting-edge knowledge and analytical skills to solve complex computing problems in industry, academia, or entrepreneurship.

PEO2: Engage in research and development activities that contribute to scientific discovery, technological advancement, or innovative applications of computing in multidisciplinary domains.

PEO3: Take on leadership roles in technical teams, research groups, or professional organizations, fostering effective collaboration and promoting ethical and inclusive practices in diverse working environments.

PEO4: Pursue continued learning and professional development through advanced studies, certifications, publications, or leadership in technology-driven

initiatives, contributing to sustainable development and digital transformation at local and global levels.

1.3.2. Program Learning Outcomes

After successful completion of the program, students are able to:

Knowledge:

1. Apply advanced knowledge of computer science theory, algorithms, data structures, and computational models to analyze and solve complex computing problems in academic, industry, or interdisciplinary contexts.

Skills:

2. Formulate research questions, design experiments or computational studies, and apply rigorous methodologies to generate new knowledge, validate hypotheses, or develop novel technologies in the field of computer science.
3. Design, implement, and evaluate large-scale or innovative software systems that meet user requirements, performance constraints, and security, ethical, or environmental considerations.
4. Communicate effectively in technical, academic, and professional settings, including writing research papers, preparing technical documentation, and delivering clear oral presentations to both expert and non-expert audiences.
5. Collaborate effectively within interdisciplinary teams, demonstrating leadership, initiative, and a commitment to inclusivity, while managing complex technical projects or research initiatives.

Accountability:

6. Recognize and respond to ethical, legal, and societal implications of computing technologies, particularly in areas such as AI, data privacy, cybersecurity, and the impact of automation.

Self-Automony:

7. Demonstrate the ability to acquire and apply emerging computing knowledge independently through research, continuous professional development, or engagement in the computing community.

1.4 Job Positions for Graduates

- Senior Software Developer and Engineers
- System and Cybersecurity analysts
- Data scientists and analysts
- Digital policy advisors
- Project managers
- Research scientists in Computer Science or Information Systems
- Further study of PhD in Computer Science
- IT Manager/Director

2. ADMISSION CANDIDATES

Candidates refer to those who meet the following requirements:

a. Having graduated with a bachelor's degree with a good grade or higher, or possess scientific publications (books, textbooks, articles published in academic journals, or papers published in the proceedings of specialized conferences or symposia) related the discipline of computer science or relevant disciplines, in particular:
   - Relevant Disciplines of Computer Science: Computer Science, Computer Networks and Data Communication, Software Engineering, Information System, Computer Engineering, Artificial Intelligence, Information Technology, Information Security, Information Technology Management, Information System Management;
   - Close disciplines: Applied Mathematics; Cryptographic Engineering, Electronics and Telecommunication Engineering; Electrical Engineering; Mathematics Informatics;
   - For other disciplines, the Admissions Council will consider and decide.

   Candidates from closely related or other disciplines are required to take 4–16 credits of supplementary courses from the list below:

No	Course Code	Name of course	Number of credits
1	MATH2020	Discrete Mathematics	4
2	COMP2050	Artificial Intelligence	4
3	COMP3010	Algorithm Design	4
4	COMP3020	Machine Learning	4

b. Having an IELTS certificate of 6.5 (or equivalent) in English within two years (24 months) before the admission registration date or have a bachelor's degree in English or graduate from a bachelor's program taught in English.

3. CURRICULUM STRUCTURE

3.1 Curriculum Composition

No.	Curriculum Components	Number of Credits	Notes
I	**COURSE WORK**		
I.1	*Required courses*	***19***	
1	*Philosophy*	*3*	
2	*Research Communication*	*4*	*with PhD programs*
3	*Major course 1*	*4*	
4	*Major course 2*	*4*	
5	*Major course 3*	*4*	
I.2	*Elective courses*	***11***	
	Students select 3-4 project-based courses		
II	**RESEARCH WORK**	***30***	
1	*Research Proposal*	*5*	
2	*Research Project 1*	*5*	
3	*Research Project 2*	*5*	
4	*Master Thesis*	*15*	
	TOTAL	**60**	

3.2 Courses and Credit Distribution by Courses

No	Course code	Name of Courses	Credit	Credit Hour Allocation		Prerequisite(s)	Grading System
				Theory	Practice		
I	**COURSE WORK**		**30**				
II.1	**Required Courses**		**19**				
1	PHIL5010	Philosophy	3	3	0		
2	CECS5010	Research Communication	4	3	1		
3	COMP5310	Algorithms and Optimization	4	3	1	Introduction to Programming (Python), Algorithms and Data Structures	
4	COMP5320	Advanced Software Engineering	4	3	1	Introduction to Programming (Python), Algorithms and Data Structures	
5	COMP5330	Advanced Database Systems	4	3	1	Object-oriented Programming, Algorithms and Data Structures	
I.2	***Elective courses*** *(Students select 3-4 project-based courses from the list below or from other Master and PhD programs)*		**11**				
1	COMP5030	Computer Vision	4	3	1	Basic knowledge of probability, linear algebra, and calculus. Python programming experience and previous exposure to image processing are highly desirable.	
2	COMP5040	Natural Language Processing	4	3	1	Advanced undergraduates and graduates with a background in formal language and automata theory. Programming	

No	Course code	Name of Courses	Credit	Credit Hour Allocation		Prerequisite(s)	Grading System
				Theory	Practice		
						experience is necessary for the assignments. The required programming language for all assignments is Python. Prior exposure to linguistics is not required.	
3	COMP5140	Text & Web Analytics	4	3	1	Introduction to programming	
4	COMP5120	Visualization	4	3	1	Statistics and Probability, Python	
5	CECS5050	Robotics	4	3	1	Control Systems	
6	COMP5070	Internet of Things	4	3	1	Networks or Equivalent	
7	COMP5090	Cryptography	4	3	1	Algorithm Design or other equivalent courses	
II.	**RESEARCH WORK**		**30**				
1	COMP5960	Research Proposal	5		*5*		
2	COMP5970	Research Project 1	5		*5*		
3	COMP5980	Research Project 2	5		*5*		
4	COMP5990	Master Thesis	15		*15*		
TOTAL			**60**				

3.3 Curriculum Planner

No	Course code	Name of Courses	Credits	Semester			
				1	2	3	4
I	**COURSE WORK**		**30**				
I.1	**Required courses**		**19**				
1	PHIL5010	Philosophy	*3*	*X*			
2	CECS5010	Research Communication	*4*	*X*			
3	COMP5310	Algorithms and Optimization	4		*X*		
4	COMP5320	Advanced Software Engineering	*4*		*X*		
5	COMP5330	Advanced Database Systems	*4*	*X*			
I.2	**Elective courses** *(Students select 3-4 project-based courses from the list below or from other BSc, Master and PhD programs)*		**11**				
1	COMP5030	Computer Vision	4	*X*			
2	COMP5040	Natural Language Processing	4		*X*		
3	COMP5140	Text & Web Analytics	4		*X*		
4	COMP5120	Visualization	4	*X*			
5	COMP5050	Robotics	4		*X*		
6	COMP5070	Internet of Things	4		*X*		
7	COMP5090	Cryptography	4	*X*			
II	**RESEARCH WORK**		***30***				
1	COMP5960	*Research Proposal*	*5*	X			
2	COMP5970	*Research Project 1*	*5*		X		
3	COMP5980	*Research Project 2*	*5*			X	
4	COMP5990	*Master Thesis*	*15*				X
TOTAL:			**60**				

3.4 Course Descriptions

PHIL5010: Philosophy

3 credits

Pre-requisites: none

This course introduces fundamental knowledge of philosophy. Topics include characteristics of Western philosophy, Eastern philosophy and Marxist philosophy; advanced content on Marxist-Leninist philosophy in the current period and its role in worldview and methodology; interrelationship between philosophy and science; the role of science in social life.

CECS5010: Research Communication

4 credits

Pre-requisites: none

This course introduces and discusses practical aspects of research communication skills, including technical paper reviewing, writing and oral presentation. Students will learn about scientific publications and peer-review, and effective scientific communications through extensive practical training including written, spoken, and individual exercises drawn from their own research.

COMP5320: Advanced Software Engineering

4 credits

Pre-requisites: Introduction to Programming (Python), Algorithms and Data Structures

This course provides a comprehensive overview of software engineering, focusing on both the theoretical foundations and practical aspects of software development. It covers key topics such as requirements specification, software design, project management, dependable and critical systems development, verification and validation, and software evolution. Emphasis is placed on understanding and applying the concept of the software engineering process, with particular attention to system models that support effective and scalable software development. By the end of the course, students will be equipped with advanced knowledge and methodologies to design, manage, and maintain complex software systems in real-world contexts.

COMP5310: Algorithms and Optimization

4 credits

Pre-requisites: Introduction to Programming (Python), Algorithms and Data Structures

This course provides a rigorous foundation in the theory and practice of algorithms and optimization. Students will study computational complexity and algorithm analysis, explore advanced data structures such as heaps, trees, graphs, and networks, and learn fundamental algorithm design paradigms including divide-and-conquer, greedy methods, and dynamic programming. The course further covers graph algorithms and network flows, the theory of NP-completeness, hardness and approximation, mathematical programming, and combinatorial optimization techniques such as branch-and-bound. Students will also be introduced to local search and metaheuristics, as well as planning and scheduling models and algorithms. By the end of the course, they will gain both the theoretical insights and practical skills necessary to design and analyze efficient algorithms and apply optimization methods to complex real-world problems.

COMP5330: Advanced Database Systems

4 credits

Pre-requisites: Introduction to Programming (Python), Algorithms and Data Structures

This course offers an in-depth exploration of the design and internals of modern database management systems, with a particular focus on the core components that underpin large-scale analytical systems (OLAP). Students will examine fundamental concepts, architectures, and implementation techniques, emphasizing both efficiency and correctness in system design. Through this study, participants will gain a deeper understanding of how databases manage storage, indexing, query processing, transaction management, and optimization at scale. The course is intended for graduate students specializing in software systems as well as advanced undergraduates with strong systems programming experience.

COMP5030 Computer Vision

4 credits

Pre-requisites: Basic knowledge of probability, linear algebra, and calculus. Python programming experience and previous exposure to image processing are highly desirable.

Computer vision is the discipline of "teaching machines how to see". Computer vision is a subfield of artificial intelligence (AI) and machine learning (ML) that focuses on enabling (teaching) computers to extract meaningful information from visual data, such as images or videos. It involves developing algorithms and techniques to understand, interpret, and analyze visual content, simulating human visual perception. It plays a crucial role in numerous applications, including object

recognition, image and video analysis, autonomous vehicles, medical imaging, augmented reality, robotics, and so on. This course will provide a coherent perspective on the different aspects of computer vision (image formulation, image processing and low-level vision, object detection, image recognition, computer vision applications, etc.), and give students the ability to understand state-of-the-art vision literature and implement components that are fundamental to many modern vision systems.

COMP5040: Natural Language Processing

4 credits

Pre-requisites: Advanced undergraduates and graduates with a background in formal language and automata theory. Programming experience is necessary for the assignments. The required programming language for all assignments is Python. Prior exposure to linguistics is not required.

This course covers the introduction to natural language processing (NLP), the goal of which is to enable computers to use human languages as input, output, or both. It examines NLP in context of different tasks including machine translation, automatic conversational assistants, and Internet search. Possible topics include summarization, machine translation, sentiment analysis and information extraction as well as methods for handling the underlying phenomena (e.g., syntactic analysis, word sense disambiguation, discourse analysis, their shortcomings and solutions).

COMP5140 Text & Web Analytics

4 credits

Pre-requisites: Introduction to Programming

Given the dominance of text information over the Internet, mining high-quality information from text becomes increasingly critical. The actionable knowledge extracted from text data facilitates our life in a broad spectrum of areas, including business intelligence, information acquisition, social behaviour analysis and decision making. In this course, we will cover important topics in text mining including basic natural language processing techniques, document representation, text categorization and clustering, document summarization, sentiment analysis, social network and social media analysis, probabilistic topic models and text visualization.

In addition, as we are in the era of Big Data, this course will provide students with opportunities to gain hands-on experience of handling large-scale data set, i.e., Big Data. Modern data processing architecture, e.g., Apache Hadoop, Apache Spark and GraphLab, will be incorporated in homework assignments.

COMP5120 Visualization

4 credits

Pre-requisites: Introduction to Programming; Probability and Statistics

Visual media are increasingly generated, manipulated, and transmitted by computers. When well designed, such displays capitalize on human facilities for processing visual information and thereby improve comprehension, memory, inference, and decision making. Yet the digital tools for transforming data into visualizations still require low-level interaction by skilled human designers. As a result, producing effective visualizations can take hours or days and consume considerable human effort.

In this course, we will study techniques and algorithms for creating effective visualizations based on principles and techniques from graphic design, visual art, perceptual psychology, and cognitive science. The course is targeted both towards students interested in using visualization in their own work, as well as students interested in building better visualization tools and systems. In addition to participating in class discussions, students will have to complete several short programming and data analysis assignments as well as a final programming project.

COMP 5050: Robotics

4 credits

Pre-requisites: Control Systems

The course will cover: components of robotic systems; selection of coordinate frames; homogeneous transformations; solutions to kinematic equations; velocity and force/torque relations; manipulator dynamics in Lagrange's formulation; digital simulation of manipulator motion; trajectory planning; obstacle avoidance; controller design using the computed torque method; and different controllers for manipulators.

COMP5070 Internet of Things

4 credits

Pre-requisites: Networks or Equivalent

This course covers the main cybersecurity principles and technologies motivated by the evolving ecosystem of Internet of Things (IoT): smart devices, sensors, operating systems, data storage, networking, communication protocols, and system services. The topics include IoT device and system security threats, privacy issues, open challenges, and countermeasure techniques.

COMP5090 Cryptography

4 credits

Pre-requisites: Algorithm Design or Equivalent

This course explores advanced concepts and research directions in cryptography. Building upon classical cryptographic primitives, the course covers modern theoretical foundations and recent breakthroughs in secure computation, post-quantum cryptography, and privacy-enhancing technologies. Students will study security models, reductions, and advanced protocols, with emphasis on both rigorous proofs and real-world applicability.

COMP5960 Research Proposal

5 credits

Students identify a relevant and challenging research topic in computer science, conduct a comprehensive literature review, and define research questions or hypotheses. They develop a detailed research proposal outlining objectives, methodology, expected outcomes, publication plan, and timeline. The proposal must be approved by a faculty advisor and graduate research committee.

COMP5970 Research Project 1

5 credits

Students conduct a research project related to the proposed research proposal under faculty supervision. The project may involve theoretical analysis, software development, experimental work, or applied research. Deliverables include a project report and potentially a draft or submission to a Scopus-indexed publication.

COMP5980 Research Project 2

5 credits

Students conduct a research project related to the proposed research proposal under faculty supervision. The project may involve theoretical analysis, software/hardware development, experimental work, or applied research. Deliverables include a project report and potentially a draft or submission to a Scopus-indexed publication.

COMP5990 Master Thesis

15 credits

Students synthesize their research into a comprehensive thesis that demonstrates innovation, scholarly depth, and relevance to computer science. The thesis is expected to consolidate findings from the research proposal and the research projects. Students must defend the thesis before a committee and meet the graduation requirement of two Scopus-indexed publications, with at least one led by the student based on their thesis or research projects.

3 Course Outlines

Course Code	PHIL5010
Course Title	**Philosophy**
Catalogue Description	This course introduces fundamental knowledge of philosophy. Topics include characteristics of Western philosophy, Eastern philosophy and Marxist philosophy; advanced content on Marxist-Leninist philosophy in the current period and its role in worldview and methodology; interrelationship between philosophy and science; the role of science in social life.
Credit Value	3
Required or elective	Required
Pre-requisite/ Co-requisite/Exclusion	None
Textbook(s) and other required materials	- Philosophy textbook promulgated by the Ministry of Education and Training. - Reference materials under the guidance of the instructor.
Course Learning Goals	1. Fostering philosophical thinking, worldview and philosophical methodology in the fields of natural science and technology. 2. Consolidate awareness of the theoretical and philosophical basis of Vietnam's revolutionary approach, especially Vietnam's science and technology development strategy.
Course Learning Objectives	Upon completion of this course, students will be able to: 1. Understand characteristics of Western philosophy, Eastern philosophy and Marxist philosophy 2. Understand advanced content on Marxist-Leninist philosophy in the current period and its role in worldview and methodology 3. Understand the interrelationship between philosophy and science; the role of science in social life.
Topics Covered/ Indicative Syllabus	• Fundamental of philosophy o What is philosophy? o Estern vs. Western philosophy • Marxist-Leninist philosophy o The birth of Marxist-Leninist philosophy o Two basic principles of the materialist dialectic o Dialectical materialism o Historical materialism o Marxist-Leninist philosophy in the present period • The relationship between science and philosophy • The role of science in social life o Scientific Consciousness o Science and technology - the driving force of social development o Science and technology in Vietnam
Class/Laboratory Schedule	Two (2) 1.5 hour-lectures per week
Assessment Methods in	• Project (40%) • In-class participation and quizzes (20%)

Course Code	PHIL5010
Alignment with Intended Learning Outcomes	• Final exam (40%)
Course Webpage	TBA
Ethical Behavior Statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgement that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	CECS5010
Course Title	**Research Communication**
Catalogue Description	This course introduces and discusses practical aspects of research communication skills, including technical paper reviewing, writing and oral presentation, and the publication process. Students will learn about scientific publications and peer-review, and effective scientific communications through extensive practical training including written, spoken, and individual exercises drawn from their own research.
Credit Value	4
Required or elective	Required
Pre-requisite/Co-requisite/Exclusion	None
Textbook(s) and other required materials	TBD by VinUni Faculty. Sample texts include: • Paul V. Anderson (2017), *Technical Communication. A Reader-centered Approach*, 9th edition (Wadsworth).
Course Learning Goals	Students will: 1. analyze the full publication process and the structure of the best research articles in their fields; 2. learn how to write a good research article; 3. learn how to give a good oral presentation of research results.
Course Learning Objectives	Upon completion of the course, students will be able to: 1. recognise and understand the full research publication process; 2. critique the content and structure of research articles; 3. recognize and reproduce the structure of excellent research articles; 4. write a good research article that supports clarity and understanding; 5. craft and deliver an oral presentation of technical information effectively.recognize and reproduce the structure of excellent research articles; 1. organize and present data in different formats including graphs, charts, tables, etc. appropriate for various purposes; 2. write a good research article that maximizes clarity and understanding; craft and deliver an oral presentation of technical information effectively.

Topics Covered/ Indicative Syllabus	• Publication Basics
	  • Building scientific knowledge
	  • The peer-review process
	  • Research publication landscape
	  • Communication Ethics
	• Manuscript Writing
	  • Literature reading and problem finding
	  • Manuscript structure and narrative
	  • Words, sentences, and paragraphs
	  • Infographics
	  • Titles, abstracts, and cover letters
	  • Edit and revise manuscripts

Course Code	CECS5010

	• Oral Presentation
	  • Communication in an era of global science
	  • How to prepare a scientific presentation
	  • How to deliver a scientific presentation
	  • Preparing and giving conference posters
	• Public Communication

Class/Laboratory Schedule	NA
Contribution of course to meeting the professional component	PLO4 is concerned with communication in various forms, and PLO7 is concerned with continuing acquisition of knowledge. After taking this course, students should be able to produce well-written technical and research papers and deliver effective oral presentations to a variety of audiences, supporting both. Though, writing documentation will be developed in the project courses. Students will also have a better understanding of the searching and the structure research literature and its various forms. They will be able to evaluate quality of sources and locate literature so they can be more focused and discriminative in their independent research.
Assessment Methods in Alignment with Intended Learning Outcomes	• Present internet/AI resources for academic research: 20% • Present research articles in a seminar format: 20% • Write paper review: 10% • Write paper abstract and introduction: 30% • Write a short survey article: 20%
Course Webpage	TBD
Rationale for Offering	
Date/Person Prepared	July 2025, Wray Buntine based on draft from 2021
Ethical Behavior Statement	Each student in this course is expected to abide by the Vin University Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with the acknowledgment that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP5030
Course Title	**Computer Vision**
Catalogue Description	Computer vision is the discipline of "teaching machines how to see". Computer vision is a subfield of artificial intelligence (AI) and machine learning (ML) that focuses on enabling (teaching) computers to extract meaningful information from visual data, such as images or videos. It involves developing algorithms and techniques to understand, interpret, and analyze visual content, simulating human visual perception. It plays a crucial role in numerous applications, including object recognition, image and video analysis, autonomous vehicles, medical imaging, augmented reality, robotics, and so on. This course will provide a coherent perspective on the different aspects of computer vision (image formulation, image processing and low-level vision, object detection, image recognition, computer vision applications, etc.), and give students the ability to understand state-of-the-art vision literature and implement components that are fundamental to many modern vision systems.
Credit Value	4
Required or elective	Elective
Pre-requisite/ Co-requisite/ Exclusion	Basic knowledge of probability, linear algebra, and calculus. Python programming experience and previous exposure to image processing are highly desirable.
Textbook(s) and other required materials	TBD by VinUni Faculty. Sample texts include: 1. *Computer Vision: A Modern Approach* by David Forsyth and Jean Ponce (2nd Edition 2011). 2. *Computer Vision: Algorithms and Applications* by Richard Szeliski (2nd Edition 2022). 3. *Computer Vision: Models, Learning, and Inference* by Simon J.D. Prince (1st Edition 2012). Book Code SP 4. *Introductory Techniques for 3-D Computer Vision* by Trucco, Emanuele, and Alessandro Verri. Prentice Hall, 2006. 5. *Computer Vision: Principles, Algorithms, Applications, Learning*, by Davies, E. R.. Academic Press, 2018.
Course Learning Goals	By the end of this course, you will be able to: 1. Understand the fundamentals of image formation and use mathematical modelling tools to represent digital images; 2. Perform transformations and filtering operations in the time and frequency domains to achieve desired outputs such as edge detection, noise removal, line and corner detection, and image smoothing; 3. Apply morphological operations for shape recognition and template matching; 4. Know the major ideas, methods, and techniques of computer vision and pattern recognition in order to design computer vision systems; 5. Use advanced algorithms such as machine learning/deep learning techniques for object recognition and classification; 6. Be able to implement computer vision and object recognition applications; 7. Learn and understand the current state-of-the-art computer vision algorithms;

Course Code	COMP5030
Course Learning Objectives	Upon completion of the course, students will be able to: 1. Apply basic knowledge, theories and methods in image processing and computer vision to relevant problems. 2. Identify, formulate and solve problems in image processing and computer vision; 3. Design and develop practical and innovative image processing and computer vision applications or systems; 4. Survey image processing and computer vision literature to locate relevant research for a particula application; 5. Communicate effectively and work in teams to develop a working computer vision system;
Topics Covered/ Indicative Syllabus	• Camera and optics • Image filtering • Image processing • Feature detection and matching • Image compression • Multiple views and stereo • Recognition • Segmentation • Introduction to spectral imaging • Introduction to machine learning • Applications, including for example the following; - Face detection - Face recognition - OCR - Industrial applications - Medical imaging - Image stitching
Contribution of course to meeting the professional component	This course serves as the foundational course in the computer vision field. It helps students to develop and apply computer vision techniques for solving practical problems. After this course, students will be able to choose appropriate image processing methods for image filtering, image restoration, image reconstruction, segmentation, classification and representation.
Assessment Methods in Alignment with Intended Learning Outcomes	• Midterm Exam: 25% • Final Exam: 25% • Literature review: 25% • Project: 25%
Course Webpage	TBD
Rationale for Offering	Computer Vision is an important field of Artificial Intelligence concerned with questions such as "how to extract information from image or video, and how to build a machine to see". Recent explosive growth of digital imaging technology, advanced computing, and deep learning makes the

Course Code	COMP5030
	problems of automated image interpretation even more exciting and much more relevant than ever.
Date/Person Prepared	July 2025 by Dr Pham Huy Hieu
Ethical Behavior Statement	Each student in this course is expected to abide by the Vin University Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with the acknowledgment that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP5040
Course Title	Natural Language Processing
Catalogue Description	This course covers the introduction to natural language processing (NLP), the goal of which is to enable computers to use human languages as input, output, or both. It examines NLP in context of including machine translation, automatic conversational assistants and Internet search. Possible topics include summarization, machine translation, sentiment analysis and information extraction as well as methods for handling the underlying phenomena (e.g., syntactic analysis, word sense disambiguation, discourse analysis, their shortcomings and solutions).
Credit Value	4
Required or elective	Elective
Pre-requisite/Co-requisite/Exclusion	Advanced undergraduates and graduates with a background in formal language and automata theory. Programming experience is necessary for the assignments. The required programming language for all assignments is Python. Prior exposure to linguistics is not required.
Textbook(s) and other required materials	TBD by VinUni Faculty. Sample texts include: 1. Jurafsky, Daniel, and James H. Martin. *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd ed., Prentice Hall, 2009. ◦ note, 3rd edition now available online, which is updated with newer content 2. Required readings are listed on the Lecture Notes along with the slides for each lecture.
Course Learning Goals	At the end of the course, students should be able to: 1. Understand and apply advanced, modern techniques in natural language processing including areas such as machine translation, automatic conversational assistants and term recognition. 2. Use natural language processing skills to design and apply systems incorporating natural language processing.
Course Learning Objectives	On completion of the course, the student shall be able to: 1. Analyse what makes NLP problems difficult. 2. Can illustrate how the theoretical background of natural language processing (NLP) addresses these difficulties. 3. Can rewrite some common NLP tools 4. Can conclude how some common NLP tools work, their limitations and relationships to other NLP problems. 5. Can appraise an area within NLP to establish the state of the art or how to apply it to a particular application.
Topics Covered/ Indicative Syllabus	• Introduction and overview • Language modeling and tasks • Part of Speech Tagging, Sequence Labeling, and Hidden Markov Models • Distributional similarities and vector semantics • Neural architectures for language modelling • Discourse pocessing • Dialogue systems

Course Code	COMP5040
Class/Laboratory Schedule	• Course introduction and organization, general introduction to NLP • Bag-of-words, TF-IDF, cosine similarity, language modeling/N-grams • Hidden Markov models and Maximum Entropy Markov models, Named Entity Recognition and Semantic Role Labelling • Word embeddings / Vector semantics, their use in different language modelling tasks • Recursive/Recurrent Neural Networks, their use in different language modelling tasks • Encoders-decoders, Attention • Transformer basics and training • Midterm exam • Tokenisation, Neural architectures including BERT, MBART and GPT • Training basics, Huggingface library and tools • Discourse and Conversational · Introduction to large language models • Alignment and safety, training methods • Project presentations
Contribution of course to meeting the professional component	Use and development of Large Language Models is now common for AI applications. This course gives students experience with these sorts of tools and an understanding of their functionality.
Assessment Methods in Alignment with Intended Learning Outcomes	• Midterm exam (20%). • Final exam (20%). • Research project (30%) • Literature survey (30%)
Course Webpage	TBD
Rationale for Offering	Natural Language Processing is an important field of Artificial Intelligence concerned with questions such as "how to extract information or answer questions from documents". Recent explosive growth in Large Language Models like ChatGPT, advanced computing, and deep learning makes the problems of NLP even more exciting and relevant than ever.
Date/Person Prepared	July 2025, Wray Buntine
Ethical Behavior Statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgment that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP5140
Course Title	**Text and Web Analytics**
Catalogue Description	Given the dominance of text information over the Internet, mining high-quality information from text becomes increasingly critical. The actionable knowledge extracted from text data facilitates our life in a broad spectrum of areas, including business intelligence, information acquisition, social behaviour analysis and decision making. In this course, we will cover important topics in text mining including: basic natural language processing techniques, document representation, text categorization and clustering, document summarization, sentiment analysis, social network and social media analysis, probabilistic topic models and text visualization. In addition, as we are in the era of Big Data, this course will provide students with opportunities to gain hands-on experience of handling large-scale data set, i.e., Big Data. Modern data processing architecture, e.g., Apache Hadoop, Apache Spark and GraphLab, will be incorporated in homework assignments.
Credit Value	4
Required or elective	Elective
Pre-requisite/Co-requisite/Exclusion	Introduction to Programming
Textbook(s) and other required materials	TBD by VinUni Faculty. Sample texts include: 1. Mining Text Data. Springer, 2014. 2. Jurafsky, Dan, and James H. Martin. Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Dorling Kindersley Pvt, Ltd., 2014. 3. Manning, Christopher D., et al. Introduction to information Retrieval. Cambridge University Press, 2018.
Course Learning Goals	Students will: 1. Learn different methods for information extraction and retrival of textual data 2. Learn different text mining techniques
Course Learning Objectives	Upon successful completion of this course, students will be able to: 1. use basic methods for extraction of information and retrieval of textual data 2. apply text processing techniques to prepare documents for statistical modelling 3. apply relevant machine learning models for analyzing textual data and 4. correctly interpret the results 5. use machine learning models for text prediction 6. evaluate the performance of machine learning models for textual data
Topics Covered/ Indicative Syllabus	• Introduction • Natural language processing • Document representation • Text categorization • Text clustering

Course Code	COMP5140
	• Topic modeling • Document summarization • Social media and network analysis • Sentiment analysis • Text visualization
Class/Laboratory Schedule	TBD
Assessment Methods in Alignment with Intended Learning Outcomes	• Assignments: 15% • Presentation: 15% • Project: 40% • Exam: 30%
Course Webpage	TBA
Ethical Behavior Statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgement that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP5120
Course Title	Visualization
Catalogue Description	Visual media are increasingly generated, manipulated, and transmitted by computers. When well designed, such displays capitalize on human facilities for processing visual information and thereby improve comprehension, memory, inference, and decision making. Yet the digital tools for transforming data into visualizations still require low-level interaction by skilled human designers. As a result, producing effective visualizations can take hours or days and consume considerable human effort. In this course, we will study techniques and algorithms for creating effective visualizations based on principles and techniques from graphic design, visual art, perceptual psychology, and cognitive science. The course is targeted both towards students interested in using visualization in their own work, as well as students interested in building better visualization tools and systems. In addition to participating in class discussions, students will have to complete several short programming and data analysis assignments as well as a final programming project.
Credit Value	4
Required or elective	Elective
Pre-requisite/ Co-requisite/Exclusion	Introduction to Programming; Probability and Statistics
Textbook(s) and other required materials	TBD by VinUni Faculty. Sample texts include: 1. Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen. *Ggplot2: Elegant Graphics for Data Analysis*. Springer, 3rd Edition 2. Claus O. Wilke. Fundamentals of Data Visualization. O'Rielley Media, 2019
Course Learning Goals	At the end of the course, students should be able to: 1. Be able to critique and develop interactive data visualizations for different applications, taking account of human perception. 2. Be able to navigate the data visualization literature, work with different tools, and understand different uses in different domains.
Course Learning Objectives	Upon successful completion of this course, you will be able to: 1. Identify, create, and validate interactive data visualization techniques. 2. Demonstrate fluency in visualization catalogues and literature and apply concepts gleaned from it when developing new visualizations. 3. Conduct visualization design critiques and analyze novel visualizations. 4. Apply human perceptual, physiological, and cognitive limits in the design of data visualizations. 5. Describe the main applications of data visualization in domains such as business, engineering and design, and the social and physical sciences.
Topics Covered/ Indicative Syllabus	• The Purpose of Visualization • Data Frames and Data Preparation • Text Visualization • Interactive and Dynamic Visualization • Machine Learning and Visualization

Course Code	COMP5120
	• Grammar of Graphics • Statistical Transformations, Scales, Themes • Accessible Visualization • Spatial Visualization • Times Series Data Visualization
Class/Laboratory Schedule	TBD
Contribution of course to meeting the professional component	ExCel
Assessment Methods in Alignment with Intended Learning Outcomes	• Project 1: Visualization Design and Implementation (20%) • Project 2: Visualization Design and Implementation (30%) • Literature survey (25%) • Report (25%)
Course Webpage	TBD
Rationale for Offering	Data Visualization is important for masters students in AI or computer science because it translates complex data and abstract model outputs into understandable insights. This skill is crucial for exploring datasets to uncover patterns and biases before model training. It also allows for interpreting and debugging sophisticated models, like neural networks, making their "black box" nature more transparent. Most importantly, it empowers students to effectively communicate the value and results of their AI solutions to non-technical stakeholders, turning complex findings into actionable business intelligence.
Date/Person Prepared	July 2025 by Dr Le Duy Dung
Ethical Behavior Statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgment that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP 5050
Course Title	**Robotics**
Catalogue Description	The course will cover: components of robotic systems; selection of coordinate frames; homogeneous transformations; solutions to kinematic equations; velocity and force/torque relations; manipulator dynamics in Lagrange's formulation; digital simulation of manipulator motion; trajectory planning; obstacle avoidance; controller design using the computed torque method; and different controllers for manipulators.
Credit Value	4 [Theory: 3, Practice: 1]
Required or elective	Elective
Pre-requisite/Co-requisite/Exclusion	Pre-requisites: Control Systems Co-requisites: None
Textbook(s) and other required materials	Recommended text: 1. **Craig John J**, *Introduction to Robotics: Mechanics and Control*, 4th Edition, Pearson, 2017.
Course Learning Goals	Students will: 1. Learn the Components of robotics systems 2. Learn the coordinate transformation, Selection of coordinate frames, the transformation arithmetic. 3. Understand the kinematics and trajectory planning. 4. Learn the Velocities, Forces, Torques in Joint and Base Coordinates. 5. Learn Dynamic Modelling, Lagrange's energy expressions for a manipulator 6. Learn the sensors and devices in a robotics system. 7. Design PID-controller and Force-torque control strategy
Course Learning Objectives	Upon completion of the course, students will be able to: 1. Identify the basic components of robot, select the right coordinate system for particular manipulator movement, and perform position transformation between different coordinates. 2. Characterize the kinematics in static and dynamic situations, calculate and solve the kinematics problems 3. Calculate a trajectory for the desired motion of a manipulator in multidimensional space; Calculate Forces and Torques in the Joints and Arms of the robot 4. Implement dynamic modelling, Lagrange's energy expression and equation of motion 5. Describe the sensors and devices in a robotics system and robotic applications 6. Calculate the parameters of controllers.
Topics covered/ Indicative Syllabus	1. Introduction to Robotics 2. Coordinate Transformation 3. Kinematics 4. Trajectory Planning 5. Control Techniques 6. Sensors and Devices 7. Robot Applications
Class/Laboratory Schedule	Lectures: Two 75 min lectures per week Homework: Weekly assignments Exams: One midterm exams and one final exam.

Course Code	COMP 5050
	Design Project: The course has a design project that will last throughout the semester.
Contribution of course to meeting the professional component	This course serves as a graduate course in Master in CS and Ph.D. in CS programs. It contributes to the students' understanding of a broad range of topics in robotics with an emphasis on basics of manipulators, coordinating transformation and kinematics, trajectory planning, control techniques, sensors and devices, and robot applications
Assessment Methods in Alignment with Intended Learning Outcomes	Course outcomes are primarily assessed through homework assignments, project, and exams: Homework (10%), Midterm Exam (30%), Project (20%), and Final Exam (40%).
Course Webpage	TBD by VinUni Faculty
Rationale for offering	This course provides students with both basic and advanced knowledge of robot motions including kinematics and dynamics, as well as control techniques and sensor systems. This course fits into the program mission and complements the existing program curriculum because it covers a broad range of topics in robotics with an emphasis on basics of manipulators, coordinate transformation and kinematics, trajectory planning, control techniques, sensors and devices, robot applications and economics analysis. This course is unique and does not overlap with any existing courses, thereby complementing the area/subarea offerings. This course will enhance graduates' education from the control system area and will support students' career in the control system applications, especially in robotics.
Person preparing this description and date	Do Tho Truong, 4/2020
Ethical behavior statement	Each student in this course is expected to abide by the Vin University Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with the acknowledgment that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP5070
Course Title	Internet of Things (IoT)
Catalogue Description	This course covers the main cybersecurity principles and technologies motivated by the evolving ecosystem of Internet of Things (IoT): smart devices, sensors, operating systems, data storage, networking, communication protocols, and system services. The topics include IoT device and system security threats, privacy issues, open challenges, and countermeasure techniques.
Credit Value	4
Required or elective	Elective
Pre-requisite/Co-requisite/Exclusion	Networks or Equivalent
Textbook(s) and other required materials	• *The Internet of Things*, by Samuel Greengard, 2015, ISBN: 9780262527736 • *Internet of Things: Architectures, Protocols, and Standards*, by Simon Cirani, Marco Picone, and Luca Veltri • *Internet of Things: Principles and Paradigms*, by Rajkumar Buyya and Amir Vahid Dastjerdi
Course Learning Goals	At the end of the course, students should be able to: 1. understand the core technology in IoT, including embedded systems, smart devices, communication protocols, and data processing techniques. 2. understand and explain the core IoT cybersecurity principles and technologies. 3. examine and explain security threats and data trustworthiness issues in IoT based applications.
Course Learning Objectives	On completion of the course, the student shall be able to: • understand and describe concepts in IoT technology, design principles of IoT systems, and IoT application development. • analyze IoT devices and systems from a cybersecurity perspective. • identify appropriate security and privacy solutions for IoT. • explain open challenges and issues related to IoT applications.
Topics Covered/ Indicative Syllabus	• Introduction to IoT systems: definitions, applications, and technologies. • IoT cybersecurity principles and technologies. • Security threats and techniques in IoT. • Data trustworthiness and privacy in IoT. • IoT in healthcare: Interoperability and security issues. • IoT in smart home: Security risks. • Open issues, challenges, and countermeasures.
Class/Laboratory Schedule	TBD
Contribution of course to meeting the professional component	The Internet of Things (IoT) course is essential for students in AI and Computer Science, bridging theoretical knowledge with real-world applications. IoT devices are ubiquitous data generators, providing the raw material for advanced machine learning and AI models. This course

Course Code	COMP5070
	will equip students with a foundational understanding of IoT architectures, sensor technologies, data acquisition, and edge computing, all critical for deploying intelligent systems in diverse environments. Students will learn about secure interconnected devices, enabling them to concieve practical solutions for smart cities, industrial automation, and connected health. This understanding is key to creating robust, data-driven solutions in an increasingly connected world.
Assessment Methods in Alignment with Intended Learning Outcomes	• Assignments: 10% • Presentation: 15% • Project: 25% • Midterm: 20% • Exam: 30%
Course Webpage	TBD
Rationale for Offering	
Date/Person Prepared	
Ethical Behavior Statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgment that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP5090
Course Title	Cryptography
Catalogue Description	This course explores advanced concepts and research directions in cryptography. Building upon classical cryptographic primitives, the course covers modern theoretical foundations and recent breakthroughs in secure computation, post-quantum cryptography, and privacy-enhancing technologies. Students will study security models, reductions, and advanced protocols, with emphasis on both rigorous proofs and real-world applicability.
Credit Value	4
Required or elective	Elective
Pre-requisite/Co-requisite/Exclusion	Algorithm Design or Equivalent
Textbook(s) and other required materials	• *The Internet of Things*, by Samuel Greengard, 2015, ISBN: 9780262527736 • *Internet of Things: Architectures, Protocols, and Standards*, by Simon Cirani, Marco Picone, and Luca Veltri • *Internet of Things: Principles and Paradigms*, by Rajkumar Buyya and Amir Vahid Dastjerdi

Course Code	COMP5090
Course Learning Goals	At the end of the course, students should be able to: • Understand the theoretical foundations and security models of modern cryptography. • Analyze advanced primitives and protocols such as zero-knowledge proofs, MPC, and post-quantum schemes. • Evaluate cryptographic constructions under formal adversarial models. • Apply cryptography to emerging domains like blockchain and privacy-preserving systems. • Design, implement, and critique advanced cryptographic algorithms and research papers.
Course Learning Objectives	On completion of the course, the student shall be able to: 1. Analyze advanced cryptographic constructions and their formal security models. 2. Apply reduction-based security proofs to evaluate the hardness assumptions behind protocols. 3. Design and implement selected advanced cryptographic primitives and protocols. 4. Critically assess the strengths and limitations of modern cryptographic techniques, especially in emerging domains such as post-quantum cryptography and decentralized systems. 5. Engage in research by exploring open problems and presenting recent research papers in cryptography.
Topics Covered/ Indicative Syllabus	• Provable security and complexity-theoretic foundations • Advanced symmetric-key and public-key cryptography (block cipher modes, lattice-based schemes, elliptic-curve cryptography) • Zero-knowledge proofs and succinct arguments (SNARKs, STARKs) • Secure multi-party computation (MPC) and homomorphic encryption • Functional encryption and attribute-based encryption • Post-quantum cryptography (lattice-based, code-based, multivariate cryptography) • Blockchain and cryptographic consensus protocols • Cryptography for privacy (differential privacy, anonymous credentials, mixnets)
Class/Laboratory Schedule	TBD
Contribution of course to meeting the professional component	The Cryptography course equips students with advanced knowledge of algorithms, security models, and cryptographic protocols, enabling them to design, analyze, and evaluate secure systems. Through research projects, technical reports, and presentations, students strengthen problem-solving, communication, and teamwork skills while engaging with cutting-edge developments such as post-quantum cryptography and zero-knowledge proofs. The course also fosters ethical awareness of data privacy and cybersecurity issues, and builds self-directed learning habits for continuous professional growth.
Assessment Methods in Alignment with Intended Learning Outcomes	• Assignments: 10% • Presentation: 15% • Project: 25% • Midterm: 20%

Course Code	COMP5090
	• Exam: 30%
Course Webpage	TBD
Rationale for Offering	
Date/Person Prepared	Dr. Le Duy Dung, Associate Program Director of Computer Science, VinUniversity
Ethical Behavior Statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgment that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	COMP5320
Course Title	Advanced Software Engineering
Catalogue Description	This course provides a comprehensive overview of software engineering, focusing on both the theoretical foundations and practical aspects of software development. It covers key topics such as requirements specification, software design, project management, dependable and critical systems development, verification and validation, and software evolution. Emphasis is placed on understanding and applying the concept of the software engineering process, with particular attention to system models that support effective and scalable software development. By the end of the course, students will be equipped with advanced knowledge and methodologies to design, manage, and maintain complex software systems in real-world contexts.
Credit Value	4
Required or elective	Required
Pre-requisite/ Co-requisite/ Exclusion	Introduction to Programming (Python), Algorithms and Data Structures
Textbook(s) and other required materials	Sample text: • Roger S. Pressman, Software Engineering: A Practitioner`s Approach
Course Learning Goals	3. Understand and apply advanced concepts and practices in modern software engineering, from requirements to deployment.

	4. Acquire skills in designing, developing, and maintaining large-scale and dependable software systems. 5. Develop expertise in software quality assurance, project management, and emerging practices in DevOps and CI/CD. 6. Build the ability to critically analyze software engineering processes and select the most appropriate methodologies for a given context.
Course Learning Objectives	By the end of this course, you will be able to: 7. Select and justify appropriate software development and maintenance processes (e.g., agile, DevOps, refactoring, regression testing). 8. Specify system requirements using semi-formal methods (e.g., UML diagrams, use-case models). 9. Apply software design techniques, including architectural styles, design patterns, and model-driven engineering. 10. Distinguish between different levels and types of testing and apply black-box, white-box, and automated testing methods. 11. Employ software project management techniques such as effort estimation, scheduling, risk management, and metrics-driven tracking. 12. Integrate modern development environments, version control, CI/CD pipelines, and automated deployment tools. 13. Apply principles of software reliability, safety, and security in the context of critical and dependable systems. 14. Communicate effectively about software development processes and present project outcomes.
Topics covered/ Indicative Syllabus	• Software Engineering Process Models: Waterfall, agile, DevOps, hybrid approaches • Requirements Engineering: elicitation, specification, modeling, validation • System Modeling and Design: UML, architectural styles, design patterns • Software Construction: abstraction, modularization, APIs, interfaces • Concurrency and Multithreading: thread safety, synchronization, patterns for parallel programming • Distributed and Networked Software: sockets, service-oriented and microservice architectures • Data Management in Software Systems: relational and NoSQL integration, persistence frameworks • Testing and Quality Assurance: unit, integration, system testing; automated frameworks; regression testing • Verification and Validation: model checking, formal methods, dependability in critical systems • Software Project Management: cost estimation, scheduling, risk analysis, project metrics • Refactoring, Maintenance, and Evolution: technical debt, continuous improvement • DevOps and CI/CD: pipeline design, containerization, infrastructure-as-code • Human Factors and Usability: designing complex user interfaces, usability evaluation

<table>
<tr><td></td><td>

- Software Security and Reliability: secure coding practices, resilience in critical systems

</td></tr>
<tr><td>Class/Laboratory Schedule</td><td>Three hours of lectures each week + 2-hour lab every week.</td></tr>
<tr><td>Contribution of course to meeting the professional component</td><td>This course is a required advanced computer science course for all students. It extends foundational programming and software design knowledge by introducing advanced engineering processes and tools necessary for large-scale, dependable systems. Students gain the ability to analyze, select, and apply appropriate methodologies, preparing them for professional practice in software development, systems design, and IT project management.</td></tr>
<tr><td>Assessment Methods in Alignment with Intended Learning Outcomes</td><td>
<table>
<tr><th>Assignment</th><th>%Weighting</th></tr>
<tr><td>Assignments (4)</td><td>10</td></tr>
<tr><td>Project Reports (3)</td><td>50</td></tr>
<tr><td>Project Peer Review (1)</td><td>10</td></tr>
<tr><td>Mid-term Exam (1)</td><td>30</td></tr>
<tr><td>Total</td><td>100%</td></tr>
</table>
</td></tr>
<tr><td>Rationale for offering</td><td>This course equips students with the advanced software engineering knowledge and practical expertise required to develop, maintain, and evolve modern software systems. It builds on earlier programming and data structures courses while focusing on scalability, dependability, and professional practices, thereby preparing students for leadership in software development projects and advanced study in computer science.</td></tr>
<tr><td>Person preparing this description and date</td><td>Dr. Le Duy Dung, Associate Program Director of Computer Science, VinUniversity</td></tr>
<tr><td>Ethical behavior statement</td><td>Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student’s own work (with acknowledgement that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.</td></tr>
</table>

Course Code	**COMP5310**
Course Title	**Algorithms and Optimization**
Catalogue Description	This course provides a rigorous foundation in the theory and practice of algorithms and optimization. Students will study computational complexity and algorithm analysis, explore advanced data structures such as heaps, trees, graphs, and networks, and learn fundamental algorithm design paradigms including divide-and-conquer, greedy methods, and dynamic programming. The course further covers graph algorithms and network flows, the theory of NP-completeness, hardness and approximation, mathematical programming, and combinatorial optimization techniques such as branch-and-bound. Students will also be introduced to local search and metaheuristics, as well as planning and scheduling models and algorithms. By the end of the course, they will gain both the theoretical insights and practical skills necessary to design and analyze efficient algorithms and apply optimization methods to complex real-world problems.
Credit Value	4
Required or elective	Required
Pre-requisite/Co-requisite/Exclusio	Introduction to Programming (Python), Algorithms and Data Structures
Textbook(s) and other required materials	• Required readings: Kleinberg, J., & Tardos, E. (2006). Algorithm design. Pearson Education India. • Additional readings: Sedgewick, R., Wayne, K. (2011). Algorithms, 4th Edition. Addison-Wesley.
Course Learning Goals	1. Master advanced techniques in the design and analysis of algorithms for computing applications. 2. Acquire the advanced knowledge necessary to progress into more complex areas of computer science. 3. Develop problem-solving skills through continuous practice with conceptual and applied computing problems.
Course Learning Objectives	6. By the end of this course, students will be able to: 7. Design and analyze greedy algorithms (e.g., interval scheduling, minimizing lateness, minimum spanning tree). 8. Develop dynamic programming solutions (e.g., weighted interval scheduling, segmented least squares, sequence alignment, shortest paths in graphs). 9. Apply divide-and-conquer approaches (e.g., integer multiplication, linear-space sequence alignment, randomized mean finding). 10. Solve problems involving network flows, cuts, and related applications. 11. Evaluate NP-complete problems such as the traveling salesman problem and knapsack problem.

<table>
<tr><td></td><td>12. Analyze approximation algorithms using greedy methods, dynamic programming, pricing strategies, and linear programming.</td></tr>
<tr><td>Topics covered/ Indicative Syllabus</td><td>• Computational Complexity and Algorithm Analysis
• Data Structures: Heaps, Trees, Graphs and Networks
• Algorithm design paradigms – Divide & Conquer, Greedy, Dynamic Programming
• Graphs Algorithms and Network Flows
• Theory of NP-Completeness / Hardness / Approximation
• Mathematical Programming
• Combinatorial Optimization, Branch & Bound
• Local Search and Meta-Heuristics
• Planning and Scheduling Models and Algorithms</td></tr>
<tr><td>Class/Laboratory Schedule</td><td>Three hours of lectures each week + 2-hour lab every week.

One project.</td></tr>
<tr><td>Contribution of course to meeting the professional component</td><td>This course is a core requirement for master’s-level computer science students. It extends foundational knowledge in programming and data structures by introducing advanced techniques in algorithm design and optimization. Through this course, students gain the theoretical understanding and analytical skills needed to tackle complex computational problems, supporting future learning in advanced computer science topics and enabling the development of innovative, efficient solutions.</td></tr>
<tr><td>Assessment Methods in Alignment with Intended Learning Outcomes</td><td>
<table>
<tr><th>Assignment</th><th>Assessment Method</th><th>% Weighting</th></tr>
<tr><td>1. Project</td><td>Programming with report.</td><td>30%</td></tr>
<tr><td>2. Weekly Labs and Homework</td><td>Hands-on practice and written homework</td><td>20%</td></tr>
<tr><td>3. Mid-term Exam</td><td>Written</td><td>25%</td></tr>
<tr><td>4. Final Exam</td><td>Written</td><td>25%</td></tr>
<tr><td colspan="2">Total</td><td>100%</td></tr>
</table>
</td></tr>
<tr><td>Rationale for offering</td><td>A strong foundation in algorithms and optimization is essential for all advanced areas of computer science. This course builds upon introductory programming, algorithms, and data structures, equipping students with the knowledge and skills to analyze, design, and apply efficient algorithms to challenging problems in computing and beyond.</td></tr>
<tr><td>Person preparing this description and date</td><td>Dr. Le Duy Dung, Associate Program Director of Computer Science, VinUniversity</td></tr>
</table>

Ethical behavior statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgement that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.

Course Code	**COMP5330**
Course Title	**Advanced Database Systems**
Catalogue Description	This course offers an in-depth exploration of the design and internals of modern database management systems, with a particular focus on the core components that underpin large-scale analytical systems (OLAP). Students will examine fundamental concepts, architectures, and implementation techniques, emphasizing both efficiency and correctness in system design. Through this study, participants will gain a deeper understanding of how databases manage storage, indexing, query processing, transaction management, and optimization at scale. The course is intended for graduate students specializing in software systems as well as advanced undergraduates with strong systems programming experience.
Credit Value	4
Required or elective	Required
Pre-requisite/Co-requisite/Exclusion	Introduction to Programming (Python), Algorithms and Data Structures
Textbook(s) and other required materials	TBD by VinUni faculty. Sample text: ● *Database Management Systems* by Raghu Ramakrishnan and Johannes Gehrke ● *Database Management Systems* (ebook)
Course Learning Goals	1. Develop an in-depth understanding of the architecture and implementation of modern database systems. 2. Acquire the ability to analyze and optimize database performance in large-scale and distributed environments. 3. Gain hands-on experience with advanced features such as query optimization, concurrency control, recovery, and scalability. 4. Explore cutting-edge developments in cloud databases, NoSQL/NewSQL systems, and big-data analytics.
Course Learning Objectives	1. Explain and evaluate the architecture and core components of a database management system. 2. Implement and compare advanced indexing, query execution, and optimization techniques. 3. Analyze and critique concurrency control and recovery protocols in high-throughput systems. 4. Assess the design trade-offs between relational, NoSQL, and NewSQL databases for modern applications. 5. Understand distributed and parallel database systems, including fault tolerance and replication strategies. 6. Design and optimize data warehouses, OLAP systems, and streaming architectures for large-scale analytics.

<table>
<tr><td></td><td>7. Critically review current research in database system design and scalability.</td></tr>
<tr><td>Topics covered/ Indicative Syllabus</td><td>

Database System Architecture: components, buffer management, process modelsAdvanced Storage & Indexing: B+-trees, hash indexing, LSM-trees, column stores
Query Processing & Optimization: relational operators, cost-based optimization, adaptive query processing
Transactions & Concurrency Control: 2PL, MVCC, optimistic concurrency, distributed transactions
Recovery & Fault Tolerance: ARIES logging, checkpoints, replication strategies
Parallel & Distributed Databases: partitioning, distributed joins, consensus protocols (Paxos, Raft)
Data Warehousing & OLAP: star schemas, materialized views, query acceleration techniques
NoSQL & NewSQL Systems: document stores, key-value stores, graph databases, scalable SQL engines
Cloud Databases: serverless DBs, autoscaling, storage disaggregation, multi-tenant systems
Streaming & Real-Time Data Systems: architectures for continuous queries and event processing
Database Tuning & Benchmarking: workload analysis, index tuning, benchmarking frameworks
Research Directions: emerging trends in ML-enhanced query optimization, learned indexes, and AI-driven data management

</td></tr>
<tr><td>Class/Laboratory Schedule</td><td>Three hours of lectures each week + 2-hour lab every week.</td></tr>
<tr><td>Contribution of course to meeting the professional component</td><td>This course provides advanced understanding of database internals and scalability, preparing students for careers in data-intensive software systems, research, and large-scale application development. It equips graduates with the technical knowledge to evaluate and build high-performance data platforms for real-world problems.</td></tr>
<tr><td>Assessment Methods in Alignment with Intended Learning Outcomes</td><td>
<table>
<tr><th>Group Assignment</th><th>Weighting</th></tr>
<tr><td>1. ER and Query Optimization Test</td><td>25%</td></tr>
<tr><td>2. Concurrency and Recovery Test</td><td>25%</td></tr>
<tr><td>3. Assignments</td><td>20%</td></tr>
<tr><td>4. Research/Implementation Project</td><td>30%</td></tr>
<tr><td>Total</td><td>100%</td></tr>
</table>
</td></tr>
<tr><td>Rationale for offering</td><td>This course extends beyond introductory database concepts to cover advanced system-level issues, distributed databases, and large-scale analytical engines. It prepares students to critically analyze and design modern database systems that power cloud services, big-data platforms, and enterprise applications.</td></tr>
</table>

Person preparing this description and date	Dr. Le Duy Dung, Associate Program Director of Computer Science
Ethical behavior statement	Each student in this course is expected to abide by the VinUni Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work (with acknowledgement that many projects are carried out in groups in which participants will contribute equally). The Code is available on the web at (insert website). Violations are taken seriously.