

Active curriculum mapping document for MJD-CMPSC Computer Science

for 2025 (Active) unit set

MJD-CMPSC Computer Science outcomes	Unit role	identify appropriate data structures and algorithms to solve computational problems	design, implement, verify, test, and document robust and secure computer programs	apply the knowledge of design and operation of core computer system components to the development of computer systems	explain ethical, social and security aspects and constraints of contemporary computer science practice	apply team-based software development processes	communicate software and systems requirements analysis, design, implementation, testing and documentation clearly in oral and written formats
CITS1401 Computational Thinking with Python	Core	☒					
create algorithms using computational thinking to solve a range of problems							
write programs using Python 3 to implement algorithms	Core	☒	☒				☒
demonstrate the process of computational problem solving	Core		☒				☒
apply common approaches to computational problem solving	Core		☒				
CITS2211 Discrete Structures	Core						☒
demonstrate critical thinking and information literacy							
understand the mathematical foundations of computer science and the evolution and history of the discipline	Core						☒
demonstrate a familiarity with basic set theory, relations, functions, predicate logic, correctness proofs, finite state automata and automata theory as tools used in computer science and software engineering	Core		☒	☒			
identify, abstract and analyse problems in computer science and software engineering using the mathematical models	Core		☒				☒
apply logical reasoning to problems in software engineering	Core						
CITS3403 Agile Web Development	Core			☒			
explain the key components that work together to enable the delivery of data and services on today's internet							
understand the technologies behind the terms and how they fit together, and back this up with examples of hands-on programming experience	Core			☒			☒
develop web applications using client side and server side technologies.	Core		☒	☒		☒	
explain and apply agile project methodologies	Core		☒				☒

MJD-CMPSC Computer Science outcomes	Unit role	identify appropriate data structures and algorithms to solve computational problems	design, implement, verify, test, and document robust and secure computer programs	apply the knowledge of design and operation of core computer system components to the development of computer systems	explain ethical, social and security aspects and constraints of contemporary computer science practice	apply team-based software development processes	communicate software and systems requirements analysis, design, implementation, testing and documentation clearly in oral and written formats
CITS3200 Professional Computing apply computer systems knowledge and skills in project situations	Core	☒	☒	☒			☒
use agile methodology in software development	Core		☒	☒			☒
solve problems in computer systems and software development	Core		☒				
justify the professional responsibility to produce reliable software	Core				☒		
assess the social impacts of computing projects	Core				☒		
perform effectively as part of a multidisciplinary and multicultural team	Core					☒	
CITS3001 Advanced Algorithms create computer algorithms for novel problems	Core	☒					
analyse the correctness and complexity of algorithms	Core		☒				
implement algorithms in code and test them for correctness and runtime	Core	☒	☒				
evaluate and apply common algorithmic problem-solving techniques	Core						☒
evaluate and apply well-known algorithms	Core						☒
CITS2005 Object Oriented Programming implement object-oriented design of solutions to real-world problems	Core	☒	☒				
design Java programs using packages and classes with inheritance, generics, interfaces and abstract classes	Core		☒	☒			
understand the use of multithreading for designing Java programs with concurrency;	Core			☒			
make effective use of software development practices to write, test, debug and document Java programs;	Core						☒
CITS3002 Computer Networks demonstrate an understanding of the basic physical operation of networks, including the concepts of data encoding and error detection and recovery;	Core			☒			☒

MJD-CMPSC Computer Science outcomes	Unit role	identify appropriate data structures and algorithms to solve computational problems	design, implement, verify, test, and document robust and secure computer programs	apply the knowledge of design and operation of core computer system components to the development of computer systems	explain ethical, social and security aspects and constraints of contemporary computer science practice	apply team-based software development processes	communicate software and systems requirements analysis, design, implementation, testing and documentation clearly in oral and written formats
explain the design motivation for, and operation of, contemporary wide-area, local-area and wireless networking technologies;	Core		☒				☒
understand the TCP/IP protocol stack, and its support for client/server and peer-to-peer networking models;	Core			☒			☒
apply industry standard networking programming interfaces from within procedural and object-oriented programming languages;	Core		☒	☒			☒
develop distributed applications for heterogeneous computer systems; and	Core		☒	☒			
understand the security and privacy factors raised by contemporary networks and their applications.	Core		☒		☒		
CITS2002 Systems Programming	Core	☒					☒
identify and appreciate the fundamentals of the imperative programming paradigm, using the standard C programming language as an example							
decide when to choose the C programming language and its standard library for their systems programming requirements;	Core	☒					☒
apply the most appropriate techniques to successfully develop robust systems programs in the C language;	Core		☒	☒			
explain the role of an operating system in the wider computing context;	Core	☒	☒	☒			
explain the relationship and interactions between an operating system's critical components and their affect on performance;	Core	☒	☒				☒
demonstrate an understanding of the relationship between contemporary operating systems, programming languages and systems-level application programming interfaces	Core	☒	☒				☒
CITS1402 Relational Database Management Systems	Core	☒	☒	☒			
understand existing database implementation and create refinements and improvements through analysis							
understand the 'business' requirements to design a database	Core		☒				
refine the database to improve and ensure correctness and reliability	Core			☒			☒
correctly program structured query language (SQL) queries and reports	Core			☒			☒
build an application layer interface for easier user interaction	Core						☒

MJD-CMPSC Computer Science outcomes	Unit role	identify appropriate data structures and algorithms to solve computational problems	design, implement, verify, test, and document robust and secure computer programs	apply the knowledge of design and operation of core computer system components to the development of computer systems	explain ethical, social and security aspects and constraints of contemporary computer science practice	apply team-based software development processes	communicate software and systems requirements analysis, design, implementation, testing and documentation clearly in oral and written formats
CITS1003 Introduction to Cybersecurity describe the concepts of cybersecurity and information security as made up by the different domains of knowledge in the unit outline	Core			☒			
specify the role of cybersecurity risk management in an overall strategy of providing information security by applying mitigations in the different cybersecurity domains	Core			☒			
define the legal and ethical responsibilities of a cybersecurity professional	Core				☒		
CITS2200 Data Structures and Algorithms undertake problem identification via abstraction	Core	☒					☒
describe common and important data structures and algorithms in the computing discipline	Core	☒					☒
implement a range of data structures and information literacy algorithms in a high-level programming language	Core		☒	☒			
apply existing data structures and algorithms from pre-built software libraries	Core	☒	☒	☒			
design data structures and algorithms	Core	☒	☒				☒
critically assess the performance of different data structures and algorithms	Core	☒	☒				☒
CITS3011 Intelligent Agents explain various ways in which algorithms can learn, their relationships, and their potential power and pitfalls	Option						
assess the performance of algorithms and appreciate the extra requirements of algorithms operating in an autonomous context	Option						
develop and implement a software agent in a suitable software engineering framework	Option						
research a relevant area of interest and effectively communicate the results through scientific writing and experimental analysis	Option						
CITS3009 Computer Science WIL Internship apply technical, communication and interpersonal skills to a working situation	Option						

MJD-CMPSC Computer Science outcomes	Unit role	identify appropriate data structures and algorithms to solve computational problems	design, implement, verify, test, and document robust and secure computer programs	apply the knowledge of design and operation of core computer system components to the development of computer systems	explain ethical, social and security aspects and constraints of contemporary computer science practice	apply team-based software development processes	communicate software and systems requirements analysis, design, implementation, testing and documentation clearly in oral and written formats
demonstrate a high level of initiative and the ability to work independently where required	Option						
create a proposal and final report with the approval of the host supervisor	Option						
perform tasks set by an employer to a high standard	Option						
critically reflect on experience in the workplace	Option						
CITS3005 Knowledge Representation	Option						
formulate complex concepts, rules and arguments in an appropriate logical language.							
select suitable tools and formalisms to capture domain specific knowledge in a machine readable format.	Option						
describe the theoretical foundation of logic programming.	Option						
design, verify and implement logic programs.	Option						
design and implement planning systems for a given domain.	Option						
CITS3003 Graphics and Animation	Option						
demonstrate technical competence in three-dimensional graphics							
apply basic knowledge of graphics systems, graphical techniques and rendering	Option						
develop graphics applications using the OpenGL API and the C programming language	Option						
identify, formulate and create three-dimensional virtual worlds and interactively manipulate these virtual worlds	Option						
advance into areas involving graphics and spatial computation, such as games programming, computer vision and scientific visualisation	Option						
CITS3007 Secure Coding	Option						
demonstrate an understanding of the basic execution model of computer programs and how programs represent and access their resources							
explain a range of common software vulnerabilities, the threats that they pose to correct execution, and steps to mitigate them	Option						

MJD-CMPSC Computer Science outcomes	Unit role	identify appropriate data structures and algorithms to solve computational problems	design, implement, verify, test, and document robust and secure computer programs	apply the knowledge of design and operation of core computer system components to the development of computer systems	explain ethical, social and security aspects and constraints of contemporary computer science practice	apply team-based software development processes	communicate software and systems requirements analysis, design, implementation, testing and documentation clearly in oral and written formats
validate a program's input data and external connections	Option						
apply industry standard secure programming practices	Option						
develop a systems approach to the evaluation of a software system to identify security vulnerabilities	Option						
CITS3402 High Performance Computing	Option						
formulate and implement high performance computing solutions for scientific problems							
demonstrate expertise in problem solving in parallel using distributed memory and distributed shared memory architectures, the two most common frameworks for high performance computing	Option						
implement applications of high performance computing for solving algorithmic or graph problems	Option						
MATH1721 Mathematics Foundations: Methods	Bridging						
use the language of mathematics to explain mathematical concepts							
solve problems involving trigonometric, exponential and logarithmic functions	Bridging						
solve problems involving linear, quadratic and simple simultaneous equations	Bridging						
apply methods of differentiation and integration to polynomials, exponentials and logarithms	Bridging						