15 School of Computing

CO324		Computer Systems				
Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Autumn	C	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	C	15 (7.5)	50% Coursework, 50% Exam	
1	Medway	Autumn	C	15 (7.5)	80% Exam, 20% Coursework	
1	Canterbury	Spring	C	15 (7.5)	80% Exam, 20% Coursework	
1	Canterbury	Autumn	C	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours: 26
Private study hours: 124
Total study hours: 150

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 Describe the purpose of, and the interaction between, the functional hardware and software components of a typical computer system.
8.2 Identify the principal hardware and software components which enable functionality and connectivity of systems ranging in scale from the global Internet down to tiny embedded systems like those that empower the Internet of Things.
8.3 Appreciate the principles and technologies behind the Internet, including layered architectures, and how this can be used to deliver effective network services.
8.4 Describe how networks and other computer hardware interact with operating systems, and can be shared between different programs and computers.
8.5 Assess the likely environmental impact of basic decisions involving computer hardware.

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 Communicate their understanding of basic computer hardware and software. □
9.2 Develop their understanding of how network technologies underpin the Internet.
9.3 Evaluate how computer hardware and software interact to deliver functionality and services at both small and large scales.

Method of Assessment

13. Assessment methods
13.1 Main assessment methods
Canterbury and Medway
Coursework 50%
(Test) A1 In-class Test (12.5%)
(Test) A2 In-class Test (12.5%)
(Test) A3 In-class Test (12.5%)
(Test) A4 In-class Test (12.5%)
2-hour unseen examination 50%

13.2 Reassessment methods
Like for like assessment

Preliminary Reading

McLoughlin, Ian Vince (2011) Computer Architecture: an embedded approach. McGraw-Hill, 512 pp. ISBN 9780-071311-182
Tanenbaum, Andrew & Bos, Herbert (2014) Modern Operating Systems (4th Edition). Pearson Education, 1136 pp. ISBN 978-0133591-620
Kurose, James and Ross, Keith (2009) Computer networking: a top-down approach (5th Edition). Pearson Education, ISBN 978-0131365-483
Mueller, Scott (2012) Upgrading and repairing PCs (20th ed onwards). QUE Press ISBN 978-0-7897-3954-4

Pre-requisites

None

Synopsis *

This module aims to provide students with an understanding of the fundamental behaviour and components (hardware and software) of a typical computer system, and how they collaborate to manage resources and provide services in scales from small embedded devices up to the global internet. The module has two strands: 'Computer Architecture' and 'Operating Systems and Networks'. Both strands contain material which is of general interest to computer users; quite apart from their academic value, they will be useful to anyone using any modern computer system.

CO510 Software Engineering

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Whole Year	I	30 (15)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours: 70
Private study hours: 230
Total study hours: 300

Learning Outcomes

Understand the principles and practice of the development of software systems (broadly defined) – from requirements specification, design, validation, implementation, and evolution
Apply design principles and patterns while developing software systems
Create UML diagrams for modelling aspects of the domain and the software
Design and implement test plans, and apply a wide variety of testing techniques effectively and efficiently
Demonstrate the vital role of planning, documentation, estimation, quality, time, cost and risk evaluation in the business context
Show an understanding of system design, including, design simplicity, appropriateness, and styles of system thinking and focused problem solving
Show an understanding of the professional and legal duties software engineers owe to their employers, employees, customers and the wider public
Use the appropriate tools and techniques when working in groups

Method of Assessment

Main assessment methods
3-stage modelling portfolio – 10%
5-stage development in groups – 40%
Examination (2 hours) – 50%

Preliminary Reading

K. Beck. Extreme Programming Explained: Embrace Change. Addison Wesley. Upper Saddle River, NJ, USA. 2000.
G. Booch, J. Rumbaugh, I. Jacobson. The Unified Modeling Language Users Guide. Addison Wesley. 1999
G. Booch, J. Rumbaugh, I. Jacobson. The Unified Software Development Process. Addison Wesley. 1999.
P. Coad, E. Lefebvre, J. De Luca. JAVA Modeling in Color with UML: Enterprise Components and Process. Prentice Hall. 1999.
A. Cockburn. Writing Effective Use Cases. Addison-Wesley. Boston, Ma, USA. 2001.
E. M. Hall. Managing Risk: Methods for Software Systems Development. Addison-Wesley. Reading, MA, USA. 1998.
D. G. Johnson, H. Nissenbaum. Computers, Ethics and Social Values. Prentice-Hall. 1995
E. A. Kallman, J. P. Grillo. Ethical Decision Making and Information Technology: An Introduction with Cases. 3rd Edition. McGraw-Hill. 1999
D. Kulak, E. Guiney. Use Cases: Requirements in Context. Addison-Wesley. Boston, Ma, USA. 2000.
J. Newkirk, R. C. Martin. Extreme Programming in Practice. Addison Wesley. Upper Saddle River, NJ, USA. 2001.
Mauro Pezze, Michal Young. Software Testing and Analysis: Process, Principles and Techniques. John Wiley & Sons. 2007.
R. Pooley, P. Stevens. Using UML Software Engineering with Objects and Components. Addison-Wesley. 2001.
G. Schneider, J. P. Winters. Applying Use Cases: A Practical Guide. Addison-Wesley. 2001.
I. Sommerville. Software Engineering.9th Edition. Harlow, England, UK. 2010.

Pre-requisites

Pre-requisite: COMP3340: People and Computing
COMP3200: Introduction to Object-Oriented Programming
COMP5200: Further Object-Oriented Programming

Synopsis *

This module provides an introduction to basic design principles of systems, including modelling principles and the use of tools, and design patterns. It also looks into different software processes, and introduces software testing. Regarding software project management, topics All the issues cover in the module will form the basis of the group project, which entails the design, implementation and evaluation of a simple software system.
This module provides an introduction to basic design principles of systems, including modelling principles and the use of tools, and design patterns. It also looks into different software processes, and introduces software testing. Regarding software project management, topics like risk management, quality assurances are covered. Under professional practice the module covers codes of ethics and professional conduct. All the issues cover in the module will form the basis of the group project, which entails the design, implementation and evaluation of a simple software system.

CO518 Algorithms, Correctness and Efficiency

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	I	15 (7.5)	70% Exam, 30% Coursework	
1	Canterbury	Autumn	I	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours: 33
Private study hours: 117
Total study hours: 150

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 specify, test, and verify program properties [B3, B4, B5, B7 ,C1]
8.2 analyse the time and space behaviour of simple algorithms [A5, B1, B4, C2, D4];
8.3 use known algorithms to solve programming problems [A2, A4, A5, B4, D3];
8.4 make informed decisions about the most appropriate data structures and algorithms to use when designing software [A2, A4, B1, B4, C2, D3, D4, D5];

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 demonstrate an understanding of trade-offs when making design decisions [B1, C2]
9.2 make effective use of existing techniques to solve problems [B4, D3, D5];
9.3 demonstrate an understanding of how programs (can fail to) match a specification [B7, D4]
9.4 analyse and compare solutions to technical problems [B1, B4, C2, D4]

Method of Assessment

Main assessment methods
2 programming assessments (15 hours each) (25% each)
2 hour unseen written examination (50%)

Reassessment methods
Like for like.

Preliminary Reading

The Art of Computer Programming. Donald E. Knuth
The Algorithm Design Manual. Steven S. Skiena
Graphs and Applications. Joan Aldous and Robin Wilson
Graphs, Networks and Algorithms. D. Jungnickel
Data Structures and Algorithms in Java 2nd Edition. M.T. Goodrich and R. Tamassia
Algorithms and Data Structures 2nd Edition. Jeffrey H. Kingston x
The Science of Programming. David Gries

Pre-requisites

Pre-requisite: COMP5200: Further Object-Oriented Programming
COMP3250: Foundations of Computing II
COMP3830: Problem Solving with Algorithms
Pre-requisite: COMP5230: Fundamentals of Programming and Logic

Synopsis *

The curriculum covers topics in algorithms and data structures, such as hashing and graph algorithms. It addresses how to program such algorithms, as well as how to test them, reason about their correctness and analyse their efficiency. It includes a mathematical treatment of big-O notation.

CO519 Theory of Computing

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	I	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours: 32
Private study hours: 118
Total study hours: 150

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 understand specifications in formal logical notation [A5]
8.2 write formal proofs [A4,A5,C4]
8.3 understand the expressiveness of various language formalisms [B1,D4]
8.4 appreciate the difference between decidable and undecidable problems [D4]

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 Understand, use and work with formal notation of various forms [A4,A5,B3]
9.2 Understand and judge the inherent complexity of certain classes of problems, and the techniques needed to approach them [A4,A5,B1,B3,B5,C1,C4,D4,D5]

Method of Assessment

Main assessment methods
Logic and Regular Languages (Coursework) (25%)
Context-Free Languages and Decidability (Coursework) (25%)
2-hour unseen examination (50%)

Preliminary Reading

Huth, Ryan: Logic in Computer Science
Boolos, Jeffrey: Computability and Logic
Martin: Introduction to Languages and the Theory of Computation

Pre-requisites

COMP5200: Further Object-Oriented Programming
COMP3250 Foundations of Computing II

Synopsis *

Propositional & Predicate Logic, including proofs. Formal languages: finite automata, regular expressions, CFGs. Turing machines, decidability.

CO520 Further Object-Oriented Programming

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Spring	I	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	I	15 (7.5)	100% Coursework	
1	Canterbury	Spring	I	15 (7.5)	70% Exam, 30% Coursework	
1	Medway	Spring	I	15 (7.5)	100% Coursework	
1	Medway	Autumn	I	15 (7.5)	100% Coursework	
1	Canterbury	Spring	I	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours:44
Private study hours: 106
Total study hours: 150

Department Checked

Yes

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 Use advanced features of an object-oriented programming language, such as inheritance and graphical libraries, to write programs.
2 Use object-oriented analysis, design and implementation with a minimum of guidance, to recognise and solve practical programming problems involving inheritance hierarchies.
3 Design appropriate interfaces between modular components.
4 Evaluate the quality of competing solutions to programming problems.
5 Evaluate possible trade-offs between alternative solutions, for instance those involving time and space differences.
6 Thoroughly test solutions to programming problems.
7 Discuss the quality of solutions through consideration of issues such as encapsulation, cohesion and coupling.

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 Make appropriate choices when faced with trade-offs in alternative designs.
2 Recognise and be guided by social, professional and ethical issues and guidelines and the general contexts in which they apply.
3 Deploy appropriate theory and practices in their use of methods and tools.

Method of Assessment

Main assessment methods
100% Coursework

Reassessment methods
100% Coursework

Preliminary Reading

"Objects first with Java – A practical introduction using BlueJ", David J. Barnes and Michael Kölling, Pearson Education, 2017, ISBN 978-1-292-15904-1.

Pre-requisites

COMP3200: Introduction to Object-Oriented Programming

Synopsis *

This module builds on the foundation of object-oriented design and implementation found in CO320 to provide both a broader and a deeper understanding of and facility with object-oriented program design and implementation. Reinforcement of foundational material is through its use in both understanding and working with a range of fundamental data structures and algorithms. More advanced features of object-orientation, such as interface inheritance, abstract classes, nested classes, functional abstractions and exceptions are covered. These allow an application-level view of design and implementation to be explored. Throughout the course, the quality of application design and the need for a professional approach to software development is emphasised.

CO523 Fundamentals of Programming and Logic

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	I	15 (7.5)	100% Coursework	

Contact Hours

Acquisition is through two one-hour lectures per week, supported by a single one-hour supervised class per week. Self-directed learning is facilitated by directed reading and web-based material. 150 total study hours spent acquiring a practical facility and understanding of the syllabus material.

Learning Outcomes

Use advanced features of an object-oriented programming language to write programs.
Write programs with the support of an integrated development environment.
Structure data and information as class definitions.
Use object-oriented analysis, design and implementation to identify and solve practical programming problems.
Test solutions to programming problems.
Discuss the quality of solutions through consideration of issues such as encapsulation, cohesion and coupling.
Understand and apply the basic concepts of Functional Programming: how evaluation operates, how to program in a side-effect-free way, understand the role of types in general, understand the use of specific types for data modelling.
Understand the basics of Propositional and Predicate Logic: their syntax (connectives, quantifiers) and their semantics (truth tables, logical equivalences).
Design appropriate interfaces between modular components.
Evaluate the quality of competing solutions to programming problems.
Evaluate possible trade-offs between alternative solutions, for instance those involving time and space differences.
Use effectively the local network facilities and a range of software development tools, such as an integrated development environment, text editor and compiler.
Appreciate at an introductory level, the representation and structuring of information with XML as a preliminary to presentation in HTML; and also the wide range of applications of XML.

Synopsis *

The module will cover: Object-oriented programming: Fundamentals of classes and objects. Class libraries. Testing. Inheritance and polymorphism.Graphical-user interfaces (GUIs), exceptions, input-output Functional programming: Expressions, values and types. (sessions and scripts). Numbers, booleans and characters.Function definitions. Approaches to testing programs. Polymorphic types. Lists and common listprocessing functions. Tuples. Pattern matching, recursive function definitions. Library functions.Algebraic data types. Propositional Logic. Data and information representation using XML. Applications of XML.

CO527 Operating Systems and Architecture

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Spring	I	15 (7.5)	60% Exam, 40% Coursework	
1	Canterbury	Spring	I	15 (7.5)	80% Exam, 20% Coursework	

Contact Hours

Total contact hours: 24
Private study hours: 126
Total study hours: 150

Learning Outcomes

On successfully completing the module students will be able to:
Have an appreciation of modern computer architecture.
Understand the operation of computer systems, both at the hardware and software level, and understand the relationship between hardware and software within the system as a whole.
Understand the need for operating systems and be aware of their overall structure.
Be able to identify and explain issues relating to performance of systems and user programs.
Understand hardware support for high level languages and be aware of the relationship between compilers, compiled code and the operating system, and its effect on performance.
Be able to understand and modify existing operating systems as necessary. [

Method of Assessment

Main assessment methods

A1 – On line quiz (10%)
A2 – Practical class (10%)
A3 – Assessment, 10 hours (20%)
Two-hour examination (60%)

Preliminary Reading

Introduction to Operating Systems: Behind the Desktop, John English. Published by Palgrave Macmillan, 2004. ISBN 0-333-99012-9.
Structured Computer Organization (International Edition), 6th edition, Andrew S Tanenbaum and Todd Austin. Published by Pearson, 2012. ISBN 0-273-76924-3.
Applied Operating System Concepts (most variants), Abraham Silberschatz, Peter Galvin and Greg Gagne. Published by John Wiley and Sons Inc. 1999. ISBN 0-471-36508-4

Pre-requisites

Pre-requisites: COMP3240: Computer Systems or equivalent background knowledge for direct-entry students
COMP3200: Introduction to Object-Oriented Programming

Synopsis *

This module aims to provide students with a more in-depth understanding of the fundamental behaviour and components (hardware and software) of a typical computer system, and how they collaborate to manage resources and provide services. It will consider systems other than the standard PC running Windows, in order to broaden students' outlook. The module has two strands: "Operating Systems" and "Architecture", which each form around 50% of the material.

CO528 Introduction to Artificial Intelligence

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
2	Canterbury	Autumn	I	15 (7.5)	50% Coursework, 50% Exam	
1	Medway	Autumn	I	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Autumn	I	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	I	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	I	15 (7.5)	80% Exam, 20% Coursework	

Availability

Autumn or Spring

Contact Hours

Total contact hours: 22
Private study hours:128
Total study hours: 150

Department Checked

Yes

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1. Explain the motivation for designing intelligent machines, their implications and associated philosophical issues, such as the nature of intelligence and learning.
2. Describe the main kinds of state-space search algorithms, discussing their strengths and limitations.
3. Explain the main concepts and principles associated with different kinds of knowledge representation, such as logic, case-based representations, and subsymbolic/connectionist representations.
4. Explain the differences between the major kinds of machine learning problems – namely supervised learning, unsupervised learning and reinforcement learning – and describe the basic ideas of algorithms for solving those problems.
5. Describe the main concepts and principles of major kinds of biologically-inspired algorithms, and understand what is required in order to implement one such technique.
6. Describe how various intelligent-system techniques have been used in the context of several case studies, and compare different techniques in the context of those case studies.

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 Discuss and give examples of the role of analogy and metaphor in science and engineering;
2 apply mathematical and computational skills in solving problems;
3 compare different strategies for problem solving, choose a strategy and justify that choice;
4 assess the strengths and weaknesses of hypotheses and techniques;
5 use the library and appropriate internet resources in support of learning.

Method of Assessment

13.1 Main assessment methods
A1 – Practical assignment (25%)
A2 – Practical assignment (25%)
2 hour unseen written examination (50%)

13.2 Reassessment methods
Like for like

Preliminary Reading

S.J. Russell & P. Norvig, "Artificial Intelligence: a modern approach", 2nd Edition. Prentice-Hall, 2002. (main textbook)
S. Pinker. "How the Mind Works", W.W. Norton & Company, 1999.
A. Cawsey, "The Essence of Artificial Intelligence", Prentice-Hall, 1998.
P. Bentley. "Digital Biology", Simon & Schuster, 2002

Pre-requisites

COMP5200: Further Object-Oriented Programming
or COMP5230: Fundamentals of Programming and Logic
or COMP3590: Programming for Artificial Intelligence

Synopsis *

This module covers the basic principles of machine learning and the kinds of problems that can be solved by such techniques. You learn about the philosophy of AI, how knowledge is represented and algorithms to search state spaces. The module also provides an introduction to both machine learning and biologically inspired computation.

CO532 Database Systems

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Spring	I	15 (7.5)	60% Exam, 40% Coursework	
1	Canterbury	Spring	I	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	I	15 (7.5)	80% Exam, 20% Coursework	
1	Medway	Spring	I	15 (7.5)	60% Exam, 40% Coursework	
1	Medway	Spring	I	15 (7.5)	50% Coursework, 50% Exam	
1	Medway	Spring	I	15 (7.5)	80% Exam, 20% Coursework	

Contact Hours

Total contact hours: 28
Private study hours: 122
Total study hours: 150

Department Checked

Yes

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 Understand the characteristics, strengths and limitations of current database systems [A2, A5].
2 Undertake self-directed background research in the module topics [B9], synthesise information collected from a variety of sources, including other modules [B1, B3],
discuss database and data management issues with their peers and with nonspecialists [B2, D2].
3 Specify, design, implement and evaluate database solutions [C1, C2, C3], perform data manipulation and information retrieval operations [A2, C2].

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 The module will extend IT skills to cover a key area that is not addressed in other parts of their programme. In particular, students will extend their ability to make effective
use of modern information system environments. The module will also contribute to development of: self-management, adjust the pace and goals of their work to meet
deadlines [D5]; oral and written communication [D2]; Internet-based information retrieval [D3].

Method of Assessment

Main assessment methods
Coursework : 50%
2-hour unseen examination (50%)

Reassessment methods
Like for like.

Preliminary Reading

C.J. Date An Introduction to Database Systems, 8th Edition, Addison Wesley, 2004.
T M Connolly & C E Begg, Database systems : a practical approach to design, implementation and management, 6th edition, Addison Wesley, 2015
R Elmasri, M Shamkant & B Navathe, Fundamentals of database systems, 7th edition, 2017
N S Ryan & D J Smith, Database Systems Engineering, Thompson, 1995.

Pre-requisites

COMP3230: Databases and the Web
or COMP5230 Fundamentals of Programming and Logic

Synopsis *

This module provides an introduction to the theory and practice of database systems. It extends the study of information systems in Stage 1 by focusing on the design, implementation and use of database systems. Topics include database management systems architecture, data modelling and database design, query languages, recent developments and future prospects.

CO539 Web Development

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	I	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	I	15 (7.5)	50% Coursework, 50% Exam	
1	Medway	Autumn	I	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours: 26
Private study hours: 124
Total study hours: 150

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 Build and deploy highly interactive, scalable and maintainable web-based systems using various tools, platforms and frameworks.
2 Understand the technologies, and the usability and performance tradeoffs, involved in creating highly interactive web-based applications.
3 Implement simple web services and understand the relationship between web sites and web services
4 Build responsive systems for mobile devices, using the web and as applications.

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 Demonstrate comprehension of the trade-offs involved in design-choices.
2 Make effective use of IT facilities and information sources for solving problems.
3 Be able to manage their own learning and development, through self-directed study and working on continuous assessment.
4 Make effective use of a range of tools, such as a web browser and database query browser.

Method of Assessment

Main assessment methods
50% Coursework and 50% Examination

Reassessment methods
Like for like.

Preliminary Reading

Robin Nixon Learning PHP, MySQL, and JavaScript: A Step-by-Step Guide to Creating Dynamic Websites, O'Reilly, 2009
Adam Tracy, Robert Hamson, Jason Essington and Anna Tokke, GWT in Action, Manning, 2nd Edition, 2013.
Wei-Meng Lee, Beginning Android 4 Development, Wrox (Wiley), 2012

Pre-requisites

Pre-requisite (Year in Computing): COMP5830: An Introduction to Programming and Web Technologies
Co-requisite (2nd year direct entry): COMP5230: Fundamentals of Programming and Logic

Pre-requisite (all other Canterbury courses):
COMP3200: Introduction to Object-Oriented Programming
COMP3230: Databases and the Web

Synopsis *

Building scaleable web sites using client-side and and server-side frameworks (e.g. JQuery, CodeIgniter). Data transfer technologies, e.g. XML and JSON. Building highly interactive web sites using e.g. AJAX. Web services. Deploying applications and services to the web: servers, infrastructure services, and traffic and performance analysis. Web and application development for mobile devices.

CO545 Functional Programming

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
2	Canterbury	Spring	I	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	I	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours: 33
Private study hours: 117
Total study hours: 150

Department Checked

Yes

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will:

1. Have an understanding of the core concepts of functional programming: functions as the central organising principle, functions as values, side-effect free programming, evaluation via rewriting.

2. Be able to understand and use recursion, recursive data structures (e.g., lists, trees, mutually recursive data types), and higher-order functions to solve problems.

3. Be able to implement functional programs to solve problems using appropriate idioms in a modern functional programming language, including understanding the languages main idioms and basic libraries.

4. Understand the structure of a functional program from the perspective of its types, including creating their own data structures to solve problems.

5. Understand the basic theory of the lambda-calculus and its relation to modern mainstream programming languages. Understand how to read context-free descriptions of language syntax e.g. for the lambda calculus.

6. Have an understanding of the concepts of concurrent programming from a functional perspective: processes, independence, communication, and synchronisation.

7. Understand program properties exposed by function and concurrent programming: partiality, totality, side-effect freedom, non-termination, determinism, deadlock, and starvation.

Method of Assessment

Main assessment methods:
In-class test 1 – 10%
Functional Programming assessment – 15%
In-class test 2 – 10%
Concurrent Programming assessment – 15%
2 hour - Examination – 50%

Reassessment methods:
Like for like.

Preliminary Reading

"Programming in Haskell" (2nd edition), Graham Hutton, 2016.
"The Haskell Road to Logic, Maths, and Programming" Kees Doets, Jan van Eijck, 2004.
“Real World Haskell: Code You Can Believe In” Bryan O'Sullivan et al, O'Reilly Media, 2008.
“Erlang Programming”, Francesco Cesarini and Simon Thompson, O'Reilly Media, 2009.
“Programming Erlang: Software for a Concurrent World”, Joe Armstrong, Pragmatic Bookshelf, 2007.

Pre-requisites

CO3BB Problem solving with algorithms or equivalent experience.

Synopsis *

This module introduces students to the functional programming paradigm, using at least one modern functional programming language to put the core concepts into practice. The module will develop both the foundation and theory of this paradigm, as well as the practice and application of the paradigm to solve problems and build systems. The module will core topics, including:
• Functions as first-class language constructs and as a central organising principle;
• Higher-order functions and compositional programming;
• Basic semantics of functional languages;
• The role of types in programming;
• Algebraic data types and pattern matching;
• Recursion and recursive data types;
• Differences with imperative and object-oriented programming paradigms;
• Properties of programs, (e.g., purity, side-effect freedom, totality, and partiality).
• The lambda-calculus as a programming model and foundation.
• BNF grammars for representing context-free syntax, and its relation to ADTs and language manipulation.
• Testing and issues of building correct software.
The module will develop practical skills in programming and problem solving using functional programming. There will also be a chance to apply functional programming to help understand better concepts in logic and mathematics.
Later parts of the module will then consider concurrent programming in the context of functional programming, including concurrent programming models and primitives (e.g., message-passing concurrency), parallelism, synchronisation and communication, and properties of deadlock, communication-safety, and starvation.

CO600 Group Project

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
2	Canterbury	Whole Year	H	30 (15)	100% Project	
1	Canterbury	Whole Year	H	30 (15)	100% Project	
2	Medway	Whole Year	H	30 (15)	100% Project	
1	Medway	Whole Year	H	30 (15)	100% Project	
1	Canterbury	Whole Year	H	30 (15)	95% Project, 5% Coursework	

Contact Hours

Total contact hours: 28
Private study hours: 272
Total study hours: 300

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 Understand particular technical topics in depth (for instance, use of a particular programming language, or software development tool, component architecture or mathematical technique) beyond that obtainable from the rest of the programme.
8.2 demonstrate an enhanced understanding (gained from practical experience) of project organisation, implementation, analytical skills and documentation techniques (as studied in other courses).
8.3 specify, design and implement a computer-based system that meets a real need;
8.4 evaluate and choose between potential solutions to a technical problem;
8.5 evaluate and deploy appropriate tools and techniques and demonstrate a degree of innovation and/or creativity

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 Appreciate the dynamics of working in a group
9.2 Demonstrate oral presentation skills
9.3 Write a technical report
9.4 Acquire technical knowledge and understanding in an independent fashion.
9.5 Critically evaluate and reflect on work performed
9.6 Manage their time and resources effectively

Method of Assessment

Main assessment methods
Project, assessed via several deliverables including a technical report and corpus, and an individual reflective report (272 hours) (100%)

Reassessment methods
Reassessment Instrument: 100% project

Preliminary Reading

None

Pre-requisites

COMP5100: Software Engineering

Synopsis *

Students, working in small groups, undertake a project related to computer science and/or software engineering. The project may be self-proposed or may be selected from a list of project proposals. A project will involve the specification, design, implementation, documentation and demonstration of a technical artefact, demonstrating the ability to synthesise information, ideas and practices to provide a quality solution together with an evaluation of that solution.

CO620 Research Project

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Whole Year	H	30 (15)	100% Project	
1	Canterbury	Whole Year	H	30 (15)	100% Project	

Contact Hours

Total contact hours: 26
Private study hours: 274
Total study hours: 300

Department Checked

Yes

Learning Outcomes

On successfully completing the module students will be able to:
1 demonstrate an in depth understanding of particular technical topics (for instance, use of a particular programming language, or software development tool, component
architecture or mathematical technique) beyond that obtainable from the rest of the programme.
2 Apply practical and analytical skills present in the programme as a whole to a research topic that addresses a real need, and demonstrate significant innovation and/or
creativity
3 apply an appropriate research process to a substantial piece of work.

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 appreciate the open-ended nature of research problems and of effective ways of tackling such problems.
2 Demonstrate oral presentational skills
3 Write a technical report, and demonstrate the ability to synthesise information, ideas and practices to provide a quality solution
4 acquire technical knowledge and understanding in an independent fashion
5 reflect on and critically evaluate work performed.
6 Manage their time and resources effectively

Method of Assessment

Main assessment methods
Project (including 8000 word technical report, poster, presentation, and short reflective report) - 100%

Reassessment methods
100% project

Preliminary Reading

None

Pre-requisites

COMP5100: Software Engineering or COMP5521: Agile Development and Software Security A

Synopsis *

As a research project, this module is normally aimed at students who are achieving at upper second class level and above, and who may be intending to undertake research following graduation. Each student undertakes a project related to computer science and/or software engineering. The project may be self-proposed or may be selected from a list of project proposals. A project will involve background study and working on an open-ended research problem.
A small number of introductory lectures are given at the start of the project.

CO633 Computer Networks and Communications

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Spring	H	15 (7.5)	60% Exam, 40% Coursework	
1	Canterbury	Autumn	H	15 (7.5)	75% Exam, 25% Coursework	
1	Canterbury	Autumn	H	15 (7.5)	60% Exam, 40% Coursework	
1	Canterbury	Autumn	H	15 (7.5)	65% Exam, 35% Coursework	

Contact Hours

Total contact hours: 25
Private study hours: 125
Total study hours: 150

Learning Outcomes

The intended subject specific learning outcomes. On successfully completing the module students will be able to:
1 Have a comprehensive and systematic understanding of current network architectures, their individual protocol layers, including the algorithms employed, and
mechanisms used to ensure network security.
2 Be able to analyse the specification for a protocol and develop software that implements a simple protocol.
3 Be aware of performance issues in general and/or analytical terms, and of the trade-offs involved.
4 Have a deeper understanding of selected key topics at the forefront of this field, including recent developments and outstanding issues.

Method of Assessment

Main assessment methods
40% Coursework and 60% Examination

Preliminary Reading

Computer Networks (Fourth Edition), Andrew S. Tanenbaum, Prentice Hall, 2002
Data and Computer Communications (7th International Edition), William Stallings, Prentice Hall, 2004
Data Communications and Networking (3rd International Edition), Behrouz A. Forouzan, McGraw-Hill, 2003
Business Data Communications and Networking (8th International Edition), Jerry Fitzgerald and Alan Dennis, Wiley, 2004

Pre-requisites

COMP3220: Foundations of Computing I
COMP3240 Computer Systems or COMP3370 Computers and the Cloud
COMP3250 Foundations of Computing II
COMP5200 Further Object-Oriented Programming

Synopsis *

The module starts with a comprehensive and detailed study of current computer networks and communications technologies. It includes: a review of network techniques, switching and multiple access; high speed local area networks; network protocols, including data link, network, transport and application layers. A selection of key topics are looked at in greater depth to reveal the state-of-the-art and issues (problems) that remain to be solved.

CO634 Computer Security and Cryptography

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Autumn	H	15 (7.5)	70% Exam, 30% Coursework	
1	Canterbury	Autumn	H	15 (7.5)	80% Exam, 20% Coursework	
1	Canterbury	Autumn	H	15 (7.5)	70% Exam, 30% Coursework	
1	Canterbury	Spring	H	15 (7.5)	80% Exam, 20% Coursework	
1	Medway	Autumn	H	15 (7.5)	80% Exam, 20% Coursework	

1	Canterbury	Spring	H	15 (7.5)	70% Exam, 30% Coursework
1	Medway	Spring	H	15 (7.5)	80% Exam, 20% Coursework

Contact Hours

Total contact hours: 30
Private study hours: 120
Total study hours: 150

Department Checked

Yes

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 have an understanding of the algorithms used in cryptography and be able to perform implementations of selected algorithms in this area [A2][C1];
2 have an understanding of the threats faced by computer operating systems, applications and networks and the various countermeasures that can be used [A1][A3];
3 be able to make informed choices of the appropriate security measures to put into place for a given network and/or operating system [C2][B5];
4 have an understanding of how cryptography can be used for providing security within applications.

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 be able to apply relevant mathematical techniques [D4].
2 be able to analyse a problem specification and to design and implement a solution [B3][B4][D3].
3 to be aware of the relevant professional, ethical and legal issues in this subject area [B6].
4 be able to develop their own time management and organisational skills. [D5].

Method of Assessment

Main assessment methods
70% Examination and 30% Coursework

Reassessment methods
Like for like

Preliminary Reading

Charles P. Pfleeger , "Security in Computing ", 2nd ed. , September 1996, Prentice Hall William Stallings, "Cryptography and Network Security : Principles and Practice", 2nd ed. , July 1998, Prentice Hall
Rita C. Summers, "Secure Computing : Threats and Safeguards", January 1997, McGraw Hill
Bruce Schneier , "Applied Cryptography : Protocols, Algorithms, and Source Code in C", 2nd ed., December 1995, John Wiley & Sons
Jonathan Knudsen , "Java Cryptography", May 1998, O'Reilly & Associates
Scott Oaks, "Java Security", May 1998, O'Reilly & Associates
Ingemar Cox, Matthew Miller & Jeffrey Bloom, "Digital Watermarking: Principles and Practice", 2003, Morgan Kaufman.

Pre-requisites

Pre-requisite:
COMP3240 Computer Systems or COMP3370 Computers and the Cloud
COMP5270 Operating Systems and Architecture or COMP5570 Computer Systems
Module not to be taken by students having already taken CO558 Introduction to Cyber Security

Synopsis *

Security has always been an important aspect of computing systems but its importance has increased greatly in recent years. In this module you learn about areas where security is of major importance and the techniques used to secure them. The areas you look at include computer operating systems (and increasingly, distributed operating systems), distributed applications (such as electronic commerce over the Internet) and embedded systems (ranging from smart cards and pay-TV to large industrial plant and telecommunications systems).

CO636 Cognitive Neural Networks

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Autumn	H	15 (7.5)	80% Exam, 20% Coursework	
1	Canterbury	Autumn	H	15 (7.5)	80% Exam, 20% Coursework	

Contact Hours

Total contact hours: 38
Private study hours: 112
Total study hours: 150

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 Describe what is meant by neural networks, list a number of types of network and give a brief description of each together with some examples of their (actual or
potential) applications.
2 Select the appropriate neural network paradigm for a particular problem and be able to justify this choice based on knowledge of the properties and potential of this
paradigm. To be able to compare the general capabilities of a number of such paradigms and give an overview of their comparative strengths and weaknesses.
3 Explain the mathematical equations that underlie neural networks, both the equations that define activation transfer and those that define learning.
4 Analyse cognitive and neurobiological phenomena from the point of view of their being computational systems. To be able to take these phenomena and identify the
features which are important for computational problem solving.
5 Build neural networks using state of the art simulation technology and apply these networks to the solution of problems. In particular, to select from the canon of learning
algorithms which is appropriate for a particular problem domain.
6 Discuss examples of computation applied to neurobiology and cognitive psychology, both in the instrumental sense of the application of computers in modelling and in the
sense of using computational concepts as a way of understanding how biological and cognitive systems function. To be able to analyse related systems not directly
studied in the course in a similar fashion.
7 Discuss examples of neural networks as applied to neurobiology.

Method of Assessment

13.1 Main assessment methods
20% Coursework and 80% Examination

Two Simulations assessments, 12 hours total (20%)
Examination, 2 hours (80%)

13.2 Reassessment methods
Like for like.

Preliminary Reading

R.C. O'Reilly and Y. Munakata "Computational Explorations in Cognitive Neuroscience, Understanding the Mind by Simulating the Brain" A Bradford Book, MIT Press 2000
D.E. Rumelhart, J.L. McClelland and the PDP Research Group "Parallel Distributed Processing, Volume 1: Foundations" MIT Press 1986
D.E. Rumelhart, J.L. McClelland and the PDP Research Group "Parallel Distributed Processing, Volume 2: Psychological and Biological Models" MIT Press 1986
W. Bechtel and A. Abrahamson "Connectionism and the Mind, Parallel Processing Dynamics and Evolution of Networks" Blackwell Publishers 2002
S. Haykin "Neural Networks, A Comprehensive Foundation" Prentice Hall International Edition 1999
C.M. Bishop "Neural Networks for Pattern Recognition" Oxford University Press 1995
R. Ellis and G. Humphreys "Connectionist Psychology, A Text with Readings" Psyhology Press Publishers 1999

Pre-requisites

COMP3220: Foundations of Computing I
or A-level Maths or Equivalent

Synopsis *

In this module you learn what is meant by neural networks and how to explain the mathematical equations that underlie them. You also build neural networks using state of the art simulation technology and apply these networks to the solution of problems. In addition, the module discusses examples of computation applied to neurobiology and cognitive psychology.

CO637 Natural Computation

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	H	15 (7.5)	60% Exam, 40% Coursework	
1	Canterbury	Spring	H	15 (7.5)	60% Exam, 40% Coursework	

Contact Hours

Total contact hours: 22
Private study hours: 128
Total study hours: 150

Learning Outcomes

The intended subject specific learning outcomes.
1 To be able to describe what is meant by a natural computation paradigm, list a number of natural computing paradigms and give a brief description of each together with
some examples of their (actual or potential) applications.
2 To be able to select the appropriate technique for a particular problem from a set of problem-solving heuristics based on these natural computing paradigms, and to be
able to justify this choice based on a knowledge of the properties and potential of these methods. To be able to compare the general capabilities of a number of such
methods and give an overview of their comparative strengths and weaknesses.
3 To be able to analyse phenomena from the natural world from the point of view of their being computational systems. To be able to take these phenomena and distinguish
between the features which are important for computational problem solving and those that are merely a fact of their realization in the natural world.
4 To be able to implement a natural computation system on the computer, and apply this program to the solution of problems.
5 To be able to exploit library and online resources to support investigations into these areas.

Method of Assessment

Main assessment methods
40% Coursework and 60% Examination

One computational exercise (about 15 hours) (20%)
One short essay (about 1,000 words) (20%)
Examination (60%)

Preliminary Reading

Eiben, AE, Smith, JE. (2015) Introduction to Evolutionary Computing, 2nd Edition. Springer.
Dorigo, M. and Stutzle, T. (2004) Ant Colony Optimization, MIT Press.
Barnes, DJ, Chu, D. (2010) Introduction to Modeling for Biosciences, Springer

Pre-requisites

Pre-requisite: COMP3220: Foundations of Computing I
COMP3250 Foundations of Computing II
or A-level Maths or Equivalent

Pre-requisite: COMP5200 Further Object-Oriented Programming
or COMP5230 Fundamentals of Programming and Logic
or COMP3590 Programming for Artificial Intelligence

Synopsis *

There is an increasing use of nature-inspired computational techniques in computer science. These include the use of biology as a source of inspiration for solving computational problems, such as developments in evolutionary algorithms and swarm intelligence. It is therefore proposed to allow students the opportunity to become exposed to these types of methods for use in their late careers.

CO641 Computer Graphics and Animation

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Spring	H	15 (7.5)	60% Exam, 40% Coursework	
1	Canterbury	Spring	H	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	H	15 (7.5)	75% Exam, 25% Coursework	

Contact Hours

Total contact hours: 30
Private study hours: 120
Total study hours: 150

Learning Outcomes

Demonstrate a broad and systematic understanding of computer graphics, animation and digital imaging from the perspective of computing, including an appreciation of technical and artistic applications;
Demonstrate familiarity with a range of technologies, techniques and algorithms for the acquisition, generation, manipulation, presentation, storage and communication of various kinds of visual data;
Apply this knowledge, including procedural techniques, through the use of 3D modelling tools.

Method of Assessment

Modelling with Blender (44%)
Computer Graphics Quiz 2%
Computer Animation Quiz 2%
Digital Imaging Quiz 2%

2 hour unseen written examination (50%)

Preliminary Reading

Computer Graphics with OpenGL (Fourth Edition, International Edition), Donald Hearn, M. Pauline Baker and Warren Carithers, Pearson Education, 2010
Computer Animation: Algorithms and Techniques (Third Edition), Richard Parent, Morgan Kaufmann, 2013
The Complete Guide to Blender Graphics: Computer Modeling and Animation (4th Edition), John M. Blain, CRC Press, 2017

Pre-requisites

None

Synopsis *

Computer graphics and animation are important for a variety of technical and artistic applications including web design, HCI and GUI development, games and simulations, digital photography and cinema, medical and scientific visualization, etc.
This module introduces the subject from the perspective of computing. You will learn about technologies and techniques for modelling, manipulating, capturing, displaying and storing visual scenes, digital images, animations and video. You will also gain practical experience of 3D modelling tools.

CO643 Computing Law and Professional Responsibility

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Spring	H	15 (7.5)	50% Coursework, 50% Exam	
1	Medway	Spring	H	15 (7.5)	75% Exam, 25% Coursework	
1	Canterbury	Autumn	H	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	H	15 (7.5)	75% Exam, 25% Coursework	
1	Canterbury	Spring	H	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours (lectures and seminars): 30 hours
Private study hours: 120 hours
Total study hours: 150 hours

Learning Outcomes

The intended subject specific learning outcomes. On successfully completing the module students will be able to:
1 Possess an ability to appropriately understand and adopt techniques based on professional codes of practice and codes of practice of accrediting institutions.
2 Develop informed judgments about when specific actions are ethical and when legal.
3 Have a systematic understanding of basic legal processes.
4 Assess the impact of data privacy legislation on institutional activities, as well as being able to explain the basic aspects of computer law as it currently applies to IT in the UK.
5 Critically evaluate IT related health and safety issues.
6 Possess critical awareness of some of the potential legal and ethical problems of large IT projects.

The intended generic learning outcomes. On successfully completing the module students will be able to:
1 Evaluate systems in terms of health and safety issues and legal responsibilities, with an awareness of possible trade-offs.
2 Communicate legal and professional issues clearly to specialist audiences.
3 Apply relevant codes of conduct and practice.
4 Have demonstrated a commitment to being a responsible member of the seminars, forum, groups and teams that they have participated in during the module.

Method of Assessment

Main assessment methods:
Coursework (50%), Examination (50%)

Preliminary Reading

Ayres R. (1999). The Essence of Professional Issues in Computing. Prentice Hall Europe.
Bynum, TW and Rogerson, S. (2006). Computer Ethics and Professional Responsibility. Blackwell Publishing.
Dejoie, R et al. (1991). Ethical Issues in Information Systems. Boyd & Fraser.
Bott F et al. (2001). Professional Issues in Software Engineering, 3rd Edition (ISBN 0748409513). Taylor & Francis.
Langford, D. (2000). Internet Ethics. Pearson/MacMillan.
Spinello, Richard A. (2003) Case studies in information technology ethics and policy, 2nd Edition, Prentice Hall.

Pre-requisites

There are no specific pre-requisites but students are expected to have completed Stage 1 of a relevant programme.

Synopsis *

The scope of the module is outlined below. Note that topics will not necessarily be delivered in this order:
Professional issues and professional organisations.
Data privacy legislation, and other UK laws relating to the professional use of computer systems.
Criminal law relating to networked computer use, including new Anti-Terrorism legislation; and their application
Intellectual Property Rights, including Copyright, Patent and Contract Law.
Health & Safety issues.
Computer-based Projects, including the vendor-client relationship and professional responsibilities.

CO645 IT Consultancy Practice 2

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	H	15 (7.5)	100% Project	
1	Medway	Spring	H	15 (7.5)	100% Project	
1	Medway	Autumn	H	15 (7.5)	100% Project	
1	Canterbury	Spring	H	15 (7.5)	100% Project	

Contact Hours

Total contact hours: 10
Private study hours and consultancy work: 140
Total study hours: 150

Learning Outcomes

Students will be able to formulate and evaluate technical alternatives to meet IT requirements arising from small businesses, including projects which have a medium-scale impact on the processes of the business. This includes issues of integration with existing technology and procedures, maintenance and expansion. Wherever appropriate, this will include consideration of both proprietary and open source solutions.
Students, working under supervision, will be able to estimate proposed solutions to smallscale IT-based problems in small business situations, in respect of both time and cost.
Students will be able to present technical and commercial aspects of proposed solutions to IT-based problems to clients, using reasoned argument attuned to the client's level of technical understanding.
Students will have demonstrated an ability to work to tightly-defined cost and timescale budgets, and have gained an understanding of how to respond in a professional manner to changes in client requirements, and other eventualities that raise the prospect of budget overruns.
Students will have gained detailed practical experience in applying selected areas of computing technology to meet the requirements of small enterprises.
Students will have experience of carrying out IT project work in a framework of defined procedures and processes, be able to evaluate that framework critically, and formulate practical proposals for small-scale developments to that framework so as to achieve a dependably high-quality service in a cost-effective way.

Method of Assessment

Main assessment methods
Project Report– 100%

Preliminary Reading

The BS EN ISO9001:2000 Standard BSi, ISBN 580368378
John Locke Open Source Solutions for Small Business Problems Charles River Media 2004, ISBN 158403203
Efraim Turban et al. Electronic Commerce: a Managerial Perspective Prentice Hall 2003, ISBN 131230158
Mark Norris and Steve West eBusiness Essentials: Technology and Network Requirements for Mobile and Online Markets John Wiley 2001, ISBN 471521833
Owen Briggs et al. Cascading Style Sheets: Separating Content from Presentation APress 2004, ISBN 159059231X

Pre-requisites

None

Restrictions

Limited enrolment capacity. Spots will be assigned on the basis of academic merit and a standard interview.

Synopsis *

Students taking this module will undertake one or (typically) more assignments for the Kent IT Consultancy (KITC). Each assignment will be of one of three types:
Work on one of KITC's contracts with an external client. To the extent that client-funded work allows, every student will be given at least one assignment of this type. Wherever practical, a student will be encouraged to participate in the negotiation and pricing of contracts, under the ultimate supervision of KITC management. For each assignment, the student may work on the assignment individually or as part of a group, as directed by KITC. A contribution to the infrastructure of KITC itself.
A contribution to the infrastructure of KITC itself. These assignments work in a similar way to external assignments, but with KITC as the client.
Formulating a costed proposal for the future development of KITC, and presenting reasoned argument in support of the proposal to KITC management, as a candidate for inclusion in KITC's strategic plan for the following academic year.

CO646 Computing in the Classroom

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Spring	H	15 (7.5)	75% Project, 25% Coursework	

Contact Hours

Lectures, classes and initial training: 14 hours
Time in school: 36 hours
Private study: 100 hours
Total study hours: 150 hours

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 Ability to present technical material succinctly and clearly to a variety of audiences
8.2 Students will realise the importance of professional responsibility and will have followed professional guidelines
8.3 Understanding of the National Curriculum and the role of Computing within it.
8.4 Knowledge of the organisation within schools and the management of people within them

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 Acting within a team, maintaining effective working relationships with teachers and pupils.
9.2 Communication skills, written and oral, both one to one and with an audience. [D2]
9.3 Ability to make effective use of general IT facilities to support their activities. [D3]
9.4 Understanding the needs of individuals.
9.5 Organisational, prioritisation, time management and negotiating skills. [D5]
9.6 Self analysis and critical evaluation. [D5, D6]
9.7 Preparation of lesson plans and teaching materials.

Method of Assessment

Main assessment methods
Weekly logs (25%)
P1 Report and portfolio (35%)
P2 Project presentation (25%)
P3 Teacher Assessment (15%)

Reassessment methods
Like for like where possible. Due to the nature of this module, and safeguarding issues relating to outside contact involving interaction with children, it is not necessarily possible to retrieve credit for this module. A different module must be chosen as a replacement if the failure is related to the actual placement.

Preliminary Reading

About UAS: http://www.uas.ac.uk
About SETNET: http://www.setnet.org.uk
Information on the National Curriculum: http://www.nc.uk.net

Pre-requisites

This module has no specific prerequisites but requires a general understanding of computing technology, as would be gained by completing Stage 1 of a computing related programme.
Acceptance on the module is by interview and is subject to DBS clearance. Acceptance does not guarantee a school placement; if no suitable placement can be found students must choose a different module.

Synopsis *

Students will spend one half-day per week for ten weeks in a school with a nominated teacher. They will observe sessions taught by their designated teacher and possibly other teachers. Later they will act somewhat in the role of a teaching assistant, by helping individual pupils who are having difficulties or by working with small groups. They may take 'hotspots': brief sessions with the whole class where they explain a technical topic or talk about aspects of university life. They must keep a weekly log of their activities. Each student must also devise a special project in consultation with the teacher and with the module convener. They must then implement and evaluate the project.

CO650 IT Consultancy Project

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Whole Year	H	30 (15)	100% Project	
1	Canterbury	Whole Year	H	30 (15)	100% Project	

Contact Hours

Total contact hours: 20
Private study hours: 300
Total study hours: 300

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 Students will be able to formulate and evaluate technical alternatives to meet IT requirements arising from small businesses, including projects which have a medium- or largescale impact on the processes of the business. This includes issues of integration with existing technology and procedures, maintenance and expansion. Wherever appropriate, this will include consideration of both proprietary and open source solutions. [A4, B1, B3, B4, B8, C2]
8.2 Students will be able to estimate proposed solutions to IT-based problems in small business situations, in respect of both time and cost. Students should be able to do this under supervision for projects of up to medium scale, and with minimal guidance for smallscale projects [B1, B8, D4]
8.3 Students will be able to present technical and commercial aspects of proposed solutions to IT-based problems to clients, using reasoned argument attuned to the client's level of technical understanding. [B2, C2]
8.4 Students will have demonstrated an ability to work to tightly-defined cost and timescale budgets, and have gained an understanding of how to respond in a professional manner to changes in client requirements, and other eventualities that raise the prospect of budget overruns. [B2, B6, D2, D5]
8.5 Students will have gained detailed practical experience in applying selected areas of computing technology to meet the requirements of small enterprises. [A1, A2, A3, B5, C3]
8.6 Students will have experience of carrying out IT project work in a framework of defined procedures and processes, be able to evaluate that framework critically, and formulate practical proposals to develop that framework so as to achieve a dependably high-quality service in a cost-effective way. [B5, B6, B8, C4, D6]
8.7 Students will be able to formulate costed plans for the strategic development of an IT consultancy business, and to canvass support for such plans by reasoned argument. [B2, B8, C1, C2, C4, D2]

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 Students will be able to explore diverse sources of information to formulate and present technical alternatives to solve a given problem, and to decide between competing solutions within an identified framework of constraints, using criteria of evaluation that they have formulated. [D2, D3]
9.2 Students will have an understanding of project management in a commercial context, including the ability to assess and manage financial, organisational, and technical risks, and the need to establish and evolve a quality management system. [A4, C2, D5]
9.3 Students will appreciate how to deal with customers in a consulting role: skills required here include communication, presentation, negotiation and (where conflict arises) conflict resolution. [B2, D1, D2]
9.4 Students will be able to interact effectively within a team, recognise and support leadership provided by others, and be able to manage conflict in this context. Students will be able spontaneously to seek and make use of advice and feedback. [D1, D5, D6]
9.5 Students will be able to take responsibility for their own work, including (where applicable) leadership and mentoring provided by them to other team members, and evaluate its strengths and weaknesses. [D1, D5]
9.6 Students will be confident in the application of their own judgement, including developing their own criteria of evaluation, and be able to challenge received opinion. These capabilities will be manifest both in the students' conduct of their own project work and (where applicable) in leadership provided to other team members. [D1, D5]

Method of Assessment

13.1 Main assessment methods
Project Report – 100%

13.2 Reassessment methods
Like for like where possible. Due to the nature of this module, and the necessity of an outside client partner, it is not necessarily possible to retrieve credit for this module. In such cases, the student must select another module, and possibly transfer to another programme when this module is compulsory on their current programme.

Preliminary Reading

The BS EN ISO9001:2000 Standard BSi, ISBN 580368378
John Locke Open Source Solutions for Small Business Problems Charles River Media 2004, ISBN 158403203
Efraim Turban et al. Electronic Commerce: a Managerial Perspective Prentice Hall 2003, ISBN 131230158
Mark Norris and Steve West eBusiness Essentials: Technology and Network Requirements for Mobile and Online Markets John Wiley 2001, ISBN 471521833
Owen Briggs et al. Cascading Style Sheets: Separating Content from Presentation APress 2004, ISBN 159059231X

Pre-requisites

None

Restrictions

Limited enrolment capacity. Spots will be assigned on the basis of academic merit and a standard interview.

Synopsis *

Students taking this module will undertake two or (typically) more assignments for the Kent IT Clinic (KITC). Each assignment will be of one of three types: .
Work on one of KITC's contracts with an external client. To the extent that client-funded work allows, every student will be given at least one assignment of this type. Wherever practical, a student will be encouraged to participate in the negotiation and pricing of contracts, under the ultimate supervision of KITC management. For each assignment, the student may work on the assignment individually or as part of a group, as directed by KITC.
A contribution to the infrastructure of KITC itself. These assignments work in a similar way to external assignments, but with KITC as the client.
Formulating a costed proposal for the future development of KITC, and presenting reasoned argument in support of the proposal to KITC management, as a candidate for inclusion in KITC's strategic plan for the following academic year. Every student will have at least one assignment of this type.

CO657 Internet of Things

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	H	15 (7.5)	100% Coursework	

Contact Hours

Total contact hours: 38 hours
Private study hours: 112 hours
Total study hours: 150 hours

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 Describe the technologies used for the Internet of Things, including (passive and active) sensors, actuators, the physical communications layer, communications
protocols, programming frameworks, and an understanding of energy and bandwidth constraints.
8.2 Design and implement software for Internet of Things applications, including both low-level firmware on embedded devices and higher-level data processing for data
obtained from sensors.
8.3 Design and build a simple sensor network based on Internet of Things technology.
8.4 Discuss and make informed comments on research into, and application areas of, the Internet of Things, including an understanding of the commercial context.

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 Communicate their understanding of technical problems and their solutions
9.2 Manage their time and resources effectively.

Method of Assessment

13.1 Main assessment methods
100% coursework

Preliminary Reading

Atzori, L., Iera, A., Morabito, G. (2010) The Internet of Things: A Survey. Computer Networks, 54(15), 2787-2805.
Fell, M. (2014) Roadmap for the Emerging "Internet of Things", Carré and Strauss.
Kortuem, G et al. (2010). Smart objects as building blocks for the Internet of things. IEEE Internet Computing, 14(1):44-51.
Welbourne, E et al. (2009). Building the Internet of Things Using RFID. IEEE Internet Computing, 13(3):48—55.
Fernandes, E., Jung, J., & Prakash, A. (2016, May). Security analysis of emerging smart home applications. In Security and Privacy (SP), 2016 IEEE Symposium on (pp. 636-654). IEEE.
Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), pp.2347-2376.

Pre-requisites

Pre-requisites:
COMP3200 Introduction to Object-oriented Programming,
COMP5200 Further Object-oriented Programming,
COMP3230 Databases and the Web,
COMP3240 Computer Systems or COMP5270 Operating systems and architecture or COMP5570 Computer Systems

Synopsis *

The module will cover a mixture of theoretical and practical topics in the area of the Internet of Things (IoT), that is, the use of Internet technologies to access and interact with objects in the physical world. This will include coverage of the range of sensor and actuator devices available, ways in which they communicate and compute, methods for getting information to and from IoT-enabled devices, and ways of visualising and processing data gained from the IoT. A practical component will consist of building the hardware and software for a sensor network and a system to collect, process and visualise data from that network.

CO658 Programming Language Implementation

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Spring	H	15 (7.5)	60% Coursework, 40% Exam	

Contact Hours

Total contact hours: 35 hours
Private study hours: 115 hours
Total study hours: 150 hours

Department Checked

Yes

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 Understand how a computer program in a high-level, imperative language is translated into machine code;
2 Understand how a program is executed, including run-time system support;
3 Understand a variety of techniques that a compiler uses to improve the efficiency of its generated code;
4 Understand how to represent programs as data in a typed functional language
5 Implement basic compiler optimisation techniques;
6 Evaluate a program's performance; and
7 Work with and modify an existing code base.

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 Be able to clearly communicate the results of performance experiments;
2 Be able to manage their own learning and development, through self-directed study and working on continuous assessment.

Method of Assessment

Main assessment methods
Coursework: 30 hours (60%)
2 hour unseen written exam (40%)

Reassessment methods
Like for like.

Preliminary Reading

Aho, A., Lam, M., Sethi, R., and Ullman, J. (2007). Compilers: Principles, Techniques, and Tools 2nd ed., Prentice Hall.
Appel, A.W. (2004) Modern compiler implementation in ML, Cambridge University Press
Cooper, K., and Torczan, L. (2011). Engineering a compiler, Morgan Kaufmann.
Minsky, Y., Madhavapeddy, A., and Hickey, J (2013). Real world OCaml, O'Reilly Media.

Pre-requisites

Pre-requisites:
COMP5450 Functional and concurrent programming;
COMP5270 Operating systems and architecture or COMP5570 Computer Systems;
COMP5180 Algorithms, correctness, and efficiency

Synopsis *

A study of techniques for interpreting and compiling programming languages, implementing them in a typed functional programming language (e.g., OCaml, Haskell). The module will outline a whole compiler from source to machine code, but will focus in depth on key algorithms and techniques. Possible in-depth topics include:

• writing interpreters,
• Hindley-Milner type inference,
• register allocation,
• garbage collection,
• abstract interpretation,
• static single assignment form.

The implemented language will be based on a simple imperative (e.g., Pascal-like) language with some extensions to address advanced topics in data layout (e.g., closures, objects, pattern matching). The course will be organized around a simple, but complete, example compiler that the student will have to understand and modify.

CO659 Computational Creativity

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Medway	Spring	H	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Spring	H	15 (7.5)	50% Coursework, 50% Exam	
1	Canterbury	Autumn	H	15 (7.5)	100% Coursework	
1	Medway	Spring	H	15 (7.5)	100% Coursework	
1	Canterbury	Spring	H	15 (7.5)	100% Coursework	

Contact Hours

Total contact hours: 38
Private study hours: 112
Total study hours: 150 hours

Learning Outcomes

On successfully completing the module students will be able to:
State and compare the various definitions of computational and human creativity, to discuss the various philosophical issues relating to computational and human creativity, and to relate these to specific examples of creative software e.g. software which composes music, writes stories, or creates scientific hypotheses.
Describe a number of computational creativity systems, both standalone and collaborative, to describe the techniques used in creating them, and describe how they are used in specific examples across a number of creative domains in the arts and sciences.
Write software that implements computational creativity techniques, grounded in an understanding of research in the area, applied to a variety of domains in the arts and sciences.
Describe, employ and debate methods for evaluation of computational creativity.
Identify appropriate contexts for using computational creativity, and design an appropriate system for that context.

Method of Assessment

Presentation and Participation in Seminars, 15 hours (20%)
Practical Project, Report and Video, 35 hours (80%)

Preliminary Reading

There is no specific textbook for this module. However students will be expected to read material provided in lectures, web-based articles and classes, as well as relevant textbooks. The following list is by way of example only:

Boden, M. (1990/2003). The Creative Mind: Myths and Mechanisms. Routledge. ISBN 978-0415314534
Dartnall, T (eds). (1994). Artificial Intelligence and Creativity: An Interdisciplinary Approach. Springer, ISBN 978-0792330615
Deliege, I., and Wiggins, G. (eds) (2006). Musical Creativity: Multidisciplinary Research in Theory and Practice. Psychology Press. ISBN 978-1841695082
McCormack, J., and d'Inverno, M. (eds). (2012). Computers and Creativity. Springer. ISBN 978-3-642-31726-2
Veale, T. (2012). Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity. Bloomsbury Acad. & Prof.. ISBN 978-1441181725
Veale, T (2014). http://robotcomix.com/ Web comics that transform our understanding of Computers and Creativity,
Selected papers from special journal issues on computational creativity, 2006-present, as listed at http://computationalcreativity.net/home/resources/journals/

Pre-requisites

Pre-requisite: COMP3200 Introduction to Object-Oriented Programming;
Pre-requisite or Co-requisite: COMP5200 Further Object-Oriented Programming, or comparable programming competence (in the latter case, please contact course convenors beforehand to confirm before registering).

Synopsis *

The module aim is to give students an overview and understanding of key theoretical, practical and philosophical research and issues around computational creativity, and to give them practical experience in writing and evaluating creative software.
The following is an indicative list of topics that may be covered:
• Introduction to computational creativity
• Examples of computational creativity software e.g. musical systems, artistic systems, linguistic systems, proof generator systems, systems for 2D and 3D design.
• Evaluation of computational creativity systems (both of the quality and the creativity of systems)
• Philosophical issues concerning creativity in computers
• Comparison of computer creativity to human creativity
• Collaborative creativity between humans and computers
• Overview of recent research directions/results in computational creativity
• Practical experience in writing creative software.

CO661 Theory and Practice of Concurrency

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	H	15 (7.5)	50% Coursework, 50% Exam	

Contact Hours

Total contact hours: 40 hours
Private study hours: 110 hours
Total study hours: 150 hours

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 Have a critical understanding of the principles of concurrent programming, as well as its advantages and challenges;
2 Reason on the properties of a distributed process (e.g., safety and liveness), and compare the behaviour of different processes.
3 Design and implement processes satisfying given properties.
4 Apply the acquired knowledge to real scenarios e.g. application-level protocols, Web services.
5 Be familiar with advanced concepts of Web Services.

The intended generic learning outcomes.
On successfully completing the module students will be able to demonstrate:
1 Systematic and rigorous reasoning,
2 Application of abstract concepts to concrete scenarios,
3 Ability of presenting and discussing state of the art topics.

Method of Assessment

Main assessment methods:
Concurrent programming in Java Coding assignment (20%)
Concurrent programming in GO Coding assignment (20%)
Seminar (Presentation and oral discussion) 10%
2-hour unseen written examination (50%)

Reassessment methods:
Like for like.

Preliminary Reading

Armstrong, J, Virding, R, Williams,W.M, Wilkstrom, (1996). C. Concurrent Programming in ERLANG. Prentice-Hall
Lynch, N.A. (1996). Distributed Algorithms. Morgan Kaufmann Publishers In (Section 7)
Milner, R. (1989). Communication and Concurrency. Prentice-Hall

Pre-requisites

Pre-requisite: COMP5450 Functional and Concurrent Programming

Synopsis *

This module is aimed at introducing the principles of concurrency theory (1, 2, 3) and demonstrating how these can be applied to design and implement distributed applications (4). Advanced concepts of Web services will be studied and placed in the perspective of these principles (5, 6).
The following is an indicative list of topics:
• Message passing primitives for concurrency: synchronous versus asynchronous message passing, the actor model.
• Reasoning on processes: temporal logic, safety and liveness properties, bisimulation.
• Channel passing and mobility.
• Design and implementation of application–level protocols.
• Web services: from stateless services to distributed business processes (also known as service orchestrations).
• Transaction protocols on the Web: two-phase commit, long running transactions

CO663 Programming Languages: Applications and Design

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Spring	H	15 (7.5)	70% Coursework, 30% Exam	
1	Canterbury	Spring	H	15 (7.5)	70% Exam, 30% Coursework	

Contact Hours

Total contact hours: 32 hours
Private study hours: 118 hours
Total study hours: 150 hours

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 Describe and compare programming paradigms: declarative, functional, imperative, object-oriented.
8.2 Solve small-scale programming problems in a range of programming languages.
8.3 Compare programming languages objectively, in order to choose the right tool for a given task.
8.4 Make effective use of a range of tools, such as editors, compilers and interpreters.

9. The intended generic learning outcomes.
On successfully completing the module students will be able to:
9.1 Demonstrate comprehension of the trade-offs involved in design-choices.
9.2 Make effective use of IT facilities for solving problems.
9.3 Manage their own learning and development, through self-directed study and working on continuous assessment.

Method of Assessment

13.1 Main assessment methods

Programming Assignment (30%)
Participatory Exercises (15%)
Paper Presentation (25%)
Examination 2 hours (30%)

13.2 Reassessment methods
Like for like.

Preliminary Reading

Bird, R. (2014). Thinking Functionally with Haskell, Cambridge University Press.
Hutton, G. (2016) Programming in Haskell, 2nd edition. Cambridge University Press
Krishnamurthi, S. (2015) [available for free online]. Programming Languages: Application and Interpretation.
Lopes, C.V. (2014). Exercises in Programming Style. Chapman and Hall/CRC.
Minsky, Y., et al. (2013). Real World OCaml, O'Reilly Media.

Pre-requisites

Pre-requisites: COMP5200 Further Object-Oriented Programming; COMP5450 Functional and Concurrent Programming

Synopsis *

This module shows students what trade-offs are involved in designing a programming language, and how those trade-offs ultimately influence programmer productivity. The module starts with a quick, example-based introduction to the basics of programming languages. It then continues with a series of problems that are each solved in several programming languages. After each problem, we stop and reflect on which language features help and which hinder. Finally, towards the end of the module, several of the language features previously identified are discussed in a more general setting. Indicative examples are:
• Basics of programming languages, such as: C++, C#, Dart, Go, Haskell, Java, Javascript, MATLAB, OCaml, Pyret, Python, Scala, Swift, R, Racket, Rust.
• Problem solving, in multiple languages. The problems will involve concepts such as parsing, evaluation, trees, graphs, memoization, randomization, big data algorithms, reactive user interfaces.
• Language features: pattern matching, first order functions, polymorphism, effects, exceptions, types, algebraic data types, modules, objects, classes.

CO822 Introduction to Quantum Computing & Quantum Cryptography

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
1	Canterbury	Autumn	M	15 (7.5)	60% Exam, 40% Coursework	
1	Canterbury	Spring	M	15 (7.5)	60% Exam, 40% Coursework	

Contact Hours

Total contact hours: 22
Private study hours: 128
Total study hours: 150

Department Checked

Yes

Learning Outcomes

8. The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
8.1 Demonstrate a systematic understanding of underlying concepts, tools, and techniques relating to quantum information processing and quantum technologies.
8.2 Demonstrate a critical awareness of quantum computing algorithms and their applications.
8.3 Creatively apply the basic tools and techniques of quantum computation and quantum information in a problem-solving scenario
8.4 Properly assess the impact of quantum technologies on science, computer security, and every-day life.
8.5 Understand how different physical systems can be used, experimentally and in practice, to represent quantum information for the purposes of computation and/or cryptography

9. The intended generic learning outcomes.
On successfully completing the module students will be:
9.1 Able to exploit library and online resources to support investigations.
9.2 Able to apply mathematical techniques where appropriate.
9.3 Able to apply appropriate scientific principles and methodology.
9.4 Able to deal with highly inter-disciplinary concepts and apply the tools of diverse fields to solving problems

Method of Assessment

Main assessment methods
2 hour written exam (60%)
Two take-home exercise worksheets (20% and approximately 15 hrs each)

Reassessment methods
Like for like.

Preliminary Reading

Phillip Kaye, Raymond Laflamme, and Michele Mosca. 2007. An Introduction to Quantum Computing. Oxford University Press, Inc., New York, NY, USA. ISBN 0198570007
Scott Aaronson. Quantum Computing since Democritus. 2013. Cambridge University Press. ISBN 9780521199568
Nielsen, M.A. and Chuang, I.L. Quantum Computation and Quantum Information. 2010. Cambridge University Press. ISBN 9780511992773

Pre-requisites

CO383 Problem Solving with Algorithms
CO322 - Foundations of Computing I and
CO325 - Foundations of Computing II
or
PH311 - Mathematics I and
PH312 - Mathematics II

Synopsis *

This module will give students an overarching introduction to quantum information processing (QIP). At the end of the course the students will have a basic understanding of quantum computation, quantum communication, and quantum cryptography; as well as the implications to other fields such as computation, physics, and cybersecurity.
We will take a multi-disciplinary approach that will encourage and require students to engage in topics outside of their core discipline. The module will cover the most essential mathematical background required to understand QIP. This includes: linear algebra, basic elements of quantum theory (quantum states, evolution of closed quantum systems, Born's rule), and basic theory of computing. The module will introduce students to the following theoretical topics: quantum algorithms, quantum cryptography, quantum communication & information. The module will also address experimental quantum computation & cryptography.

CO832 Data Mining and Knowledge Discovery

Version	Campus	Term(s)	Level	Credit (ECTS)	Assessment	Convenor
2	Canterbury	Spring	M	15 (7.5)	80% Exam, 20% Coursework	

Contact Hours

Total contact hours: 22 hours
Private study hours: 128 hours
Total study hours: 150 hours

Department Checked

Yes

Learning Outcomes

The intended subject specific learning outcomes.
On successfully completing the module students will be able to:
1 Explain the differences between the major data mining tasks, in terms of their assumptions, requirement for a specific kind of data, and the different kinds of knowledge
discovered by algorithms performing different kinds of task.
2 Describe data mining algorithms for the major data mining tasks.
3 Identify which data mining task and which algorithm is the most appropriate for a given data mining project, taking into account both the nature of the data to be mined
and the goals of the user of the discovered knowledge.
4 Use a state-of-the-art data mining tool in a principled fashion, being aware of the strengths and weaknesses of the algorithms implemented in the tool.
5 Evaluate the quality of discovered knowledge, taking into account the requirements of the data mining task being solved and the goals of the user.
6 Describe the main tasks and algorithms involved in the preprocessing and postprocessing steps of the knowledge discovery process.
7 Utilize the library and exploit web sites to support investigations into these areas.

The intended generic learning outcomes.
On successfully completing the module students will be able to:
1 Understand the major kinds of data mining tasks and the main kinds of algorithms that are often used to solve these tasks.
2 Understand the strengths and weaknesses of some data mining algorithms, identifying the kind of algorithm that is most appropriate for each data mining problem.
3 Understand the process of knowledge discovery, involving not only data mining but also preprocessing and post-processing steps

Method of Assessment

Main assessment methods
20% Coursework and 80% Examination

One exercise with a data mining tool 10%
One Short Essay (about 1,000 words) 10%
Examination 80%

Reassessment methods
Like for like.

Preliminary Reading

Witten, IH, Frank, E, Hall, MA, Pal, CJ (2016). Data Mining: practical machine learning tools and techniques, 4rd edition. Morgan Kaufmann.
Tan, P-N, Steinbach, M, Karpatne, A, Kumar, V (2018) Introduction to Data Mining, Pearson, 2nd edition.

Pre-requisites

Pre-requisite: knowledge of programming such as that provided by COMP3200 Introduction to Object-Oriented Programming
COMP8710 Advanced Java for Programmers
COMP8820 Advanced Object-Oriented Programming or
COMP8210 Programming for Data Handling

Synopsis *

This module explores a range of different data mining and knowledge discovery techniques and algorithms. You learn about the strengths and weaknesses of different techniques and how to choose the most appropriate for any particular task. You use a data mining tool, and learn to evaluate the quality of discovered knowledge.