Bachelors of Computer Science Course Plan

Semester 1 Year 1

1. Programming Bootcamp 1

Course Coordinator: Edouard Amouroux
Course Coordinator Phone: -
Course Coordinator Email: edouard.amouroux4@rmit.edu.au
Course Coordinator Location: 014.08.08
Course Coordinator Availability: by appointment

Course Description
Programming Bootcamp 1 introduces programming in a bootcamp style. This course covers algorithmic development using standard control structures, fundamental concepts of programming and the creation of small to medium practical applications. In this course, you will learn the basic skills of program development by working on the production of small amounts of software (typically in Java programming language). This will include skills in testing and debugging programs, and in the use of appropriate software technologies.
This course is tightly linked to COSC2803 / COSC3056 / COSC3057 Programming Studio 1, which should be taken right after (within the same semester). Programming Studio 1 offers an extensive opportunity to apply concepts and skills learnt in this course over a realistic programming project.

2. Mathematics for Computing 1

Course Coordinator: Son Hoang Dau

Course Coordinator Phone: N/A

Course Coordinator Email: sonhoang.dau@rmit.edu.au

Course Coordinator Location: 014.11.20

Course Coordinator Availability: by appointment

Course Description

Mathematics for Computing 1 provides a foundation for Computer Science. Many other areas of Computer Science require the ability to work with concepts from discrete structures, which include topics such asset theory, integers, functions, relations, logic, proofs, and graph theory. The material in discrete structures is pervasive in the areas of data structures and algorithms but appears elsewhere in Computer Science as well.

3. Programming Studio 1

Course Coordinator: Halil Ali

Course Coordinator Phone: -

Course Coordinator Email: halil.ali@rmit.edu.au

Course Coordinator Location: -

Course Coordinator Availability: by appointment

Course Description

Programming Studio 1 is a studio-based course that focuses on the acquisition and development of fundamental technical and professional skills for computing. The studio-based approach provides an authentic problem setting. Theory and practice are blended in the application of learning centred on development.

In Programming Studio 1 students explore how to build software, through learning how to build a simple fully functional data-driven website. Students learn about fundamental concepts in User-Centred Design (UX/UI) and Database System that explore the front-end-to-back-end of software development. Students integrate theory and practice across these concepts together in the Studio Project. In a studio-long project, students put into practice the theoretical concepts through your programming skills to make a fully functional website.

This course includes a work integrated learning experience in which your knowledge and skills will be applied and assessed in a real or simulated workplace context and where feedback from industry and/ or community is integral to your experience.

Semester 2 Year 1

1. Programming Bootcamp 2

Course Coordinator: Julie Porteous
Course Coordinator Phone: +61 3 9925 3575
Course Coordinator Email: julie.porteous@rmit.edu.au
Course Coordinator Location: 014.08.07E
Course Coordinator Availability: by appointment

Course Description
Programming Bootcamp 2 builds on Programming Bootcamp 1, in a bootcamp style, i.e. in a focussed mode over a period of 5 or 6 weeks. This course covers more advanced algorithms. This course serves as a pre-requisite for more specialized courses that require programming. This course is tightly linked to Programming Studio 2 (PS2), which should be taken immediately after (i.e. within the same semester). PS2 offers extensive opportunity to apply the skills from this course in a realistic programming project.

2. Practical Statistics

Course Coordinator: Assoc. Prof. Melih Ozlen
Course Coordinator Phone: NA
Course Coordinator Email: melih.ozlen@rmit.edu.au
Course Coordinator Availability: By appointment; by email

Course Description
This course will introduce you to fundamental concepts in statistics and data analytics. You will learn data analysis and modelling, starting with descriptive statistics and data visualisation random variables, normal and binomial probability distributions. You will build upon these and learn how to perform statistical inference including hypothesis testing and confidence intervals. You will learn statistical modelling and analysis techniques including linear regression, analysis of variance categorical data analysis and nonparametric analysis. There will be an emphasis on conceptual understanding and the use of computer software for analytical computations.

3. Programming Studio 2

Course Title: Programming Studio 2
Credit Points: 24.00
Course Coordinator: Fabio Zambetta
Course Coordinator Phone: +61 (3) 99259694

Course Coordinator Email: fabio.zambetta@rmit.edu.au

Course Description
Programming Studio 2 is an intermediate studio-based course that focuses on the further acquisition and development of technical and professional skills for computing. The studio-based approach provides an authentic problem setting where theory and practice of computer architecture are blended. In this course you will develop programming principles, skills and practices for modern software development, continuing the study of design, development and testing from previous courses. These learning outcomes will be achieved by developing a medium-sized real-world application.

Year 2

1. Computing Theory

Course Title: Computing Theory
Credit Points: 12.00
Course Coordinator: Professor James Harland
Course Coordinator Phone: +61 3 9925 2045
Course Coordinator Email: james.harland@rmit.edu.au

Course Description
Computing Theory introduces you to foundational issues in computer science. The emphasis is on understanding and applying foundational concepts and techniques. You will learn and apply fundamental theories of computing to computing problems. Topics include the study of formal models of computation and computability properties, measuring time requirements for a computation approaches to difficult problems, the use of grammars to specify syntax rules.

2. Software Engineering Fundamentals

Course Title: Software Engineering Fundamentals
Credit Points: 12.00
Course Coordinator: Dr Mojtaba Shahin
Course Coordinator Phone: .
Course Coordinator Email: mojtaba.shahin@rmit.edu.au

Course Description
This course is designed to provide you opportunity to gain knowledge and skills necessary to analyse, design and implement complex software engineering projects. You should learn to analyse and design fairly complex real-life systems, working as teams. The project based

approach used requires you to review and refine your design iteratively based on regular feedback from staff. You are also made aware of current software engineering standards and processes. You are also taught to consider qualitative aspects including maintainability, extensibility, reusability and robustness in every stage of the software-engineering life-cycle. At the end of the course you should be able to combine top-down and bottom-up approaches to software design and choose the most appropriate process considering the underlying technology, project duration, the level of risks and the customer expectations.

3. Advanced Programming Techniques

Course Title: Advanced Programming Techniques
Credit Points: 12.00
Course Coordinator: Xiaodong Li
Course Coordinator Phone: -
Course Coordinator Email: xiaodong.li@rmit.edu.au

Course Description
This course teaches programming principles, skills and practices for modern software development, continuing the study of software design and development in pre-requisite courses. Concepts in this include dynamic memory management, advanced typing and polymorphism, automated build systems, defensive programming, debugging, testing, and adherence to modern coding standards and practices. This course uses the C++ programming language for the practical implementation and practice of these concepts. This course may serve as a foundation for further studies in Computer Science and Software Engineering.

4. Software Engineering: Process and Tools

Course Title: Software Engineering: Process and Tools
Credit Points: 12.00
Course Coordinator: Sebastian Rodriguez
Course Coordinator Phone: -
Course Coordinator Email: sebastian.rodriguez@rmit.edu.au

Course Description
Software Engineering is more than the programming phase of an application arena. Phases of the software engineering life cycle include specialised processes and tools to ensure that real-world projects, both large and small, are delivered in a quality manner under financial constraints using processes and tools. Invariably, the software end-product is produced by a team of software engineers, stakeholders and ancillary personnel.

This course aims to develop your knowledge of the processes and associated tools required to service the software development life cycle.

This course includes a Work Integrated Learning experience in which your knowledge and skills will be applied and assessed in a simulated workplace context and where feedback from industry representatives will be provided. The type of WIL activities involved include a substantial team project to develop a practical software product, using modern Software Engineering processes� and reflective practice, and standard industry tools for code management, testing, and continuous integration and deployment (Assessment Task 1). The team project has been evaluated by the chair of the Computer Science Industry Advisory Committee and feedback provided to ensure it reflects industry best practice. Project mentors and lab/tutors are drawn from industry and support learning and provide feedback on modern Software Engineering practice; these practitioners will also act as client�representatives and will mark the project in conjunction with academic staff.

5. Algorithms and Analysis

Course Title: Algorithms and Analysis
Credit Points: 12.00
Course Coordinator: A/Prof. Jeffrey Chan
Course Coordinator Phone: -
Course Coordinator Email: jeffrey.chan@rmit.edu.au

Course Description
The main objective of this course is for you to acquire the tools and techniques necessary to propose practical algorithmic solutions to real-world problems which still allow strong theoretical bounds on time and space usage. You will study a broad variety of important and useful algorithms and data structures in different areas of applications, and will concentrate on fundamental algorithms. You will spend a significant time on each algorithm to understand its essential characteristics and to respect its subtleties.

6. Operating System Principles

Course Title: Operating Systems Principles
Credit Points: 12.00
Course Coordinator: Dr. Qiang Fu
Course Coordinator Phone: by email
Course Coordinator Email: qiang.fu@rmit.edu.au

Course Description
This course builds on skills gained in preliminary programming courses in both C/C++ programming languages and gives you an in-depth understanding of designing and implementing operating systems. The course covers both theoretical and practical challenges encountered when designing, implementing, and using operating systems. We also evaluate

trade-offs between conflicting objectives in large scale system design. Core topics include operating system structure, hardware interfaces and abstraction, process scheduling and dispatch, threading, synchronization, interprocess communications, memory management, file systems, system security and protection, and virtualisation.�

Year 3

1. Professional Computing Practice

Course Title: Professional Computing Practice
Credit Points: 12.00
Course Coordinator: David Eccles
Course Coordinator Phone: -
Course Coordinator Email: david.eccles@rmit.edu.au

Course Description
This course is an introduction to computing ethics, law and employability. It is intended for computing students who have not studied business principles, or who have little work experience in industry. The course provides a survival kit for computing graduates entering the workforce. During this course, you should consider computing ethical issues, such as information privacy, computer crime, computer security, social networks, online identity. During this course, you should also consider the international legal frameworks available to protect software system development. This includes non-disclosure agreements, employment contracts, intellectual property law (copyright, patents, licensing, royalties), trademarks and warranty disclaimers. Additionally, you should examine the marketing of a software system development, involving SWOT analysis and action plan.�

2. Artificial Intelligence

Course Title: Artificial Intelligence
Credit Points: 12.00
Course Coordinator: Prof. Sebastian Sardina
Course Coordinator Phone: +61 3 9925 9824
Course Coordinator Email: sebastian.sardina@rmit.edu.au

Course Description
This course introduces you to the basic concepts and techniques of Artificial Intelligence (AI). AI is the sub-area of computer science devoted to creating software and hardware to get computers to do things that would be considered �intelligent� as if people did them. Artificial intelligence has had an active and exciting history and is now a reasonably mature area of computer science. Many of the research discoveries have now reached the point of industrial

application and. many companies have made and saved millions of dollars by exploiting the results of AI research. However the goal of emulating human intelligence has not been reached and many stimulating and challenging problems remain.

All serious programmers and software engineers should know about the major AI techniques, which are regarded by many the core knowledge of any Computer Science degree. This course will allow you to gain generic problem solving skills that have applicability to a wide range of real-world problems. Topics covered include search strategies for solving problems, knowledge representation, automated planning, intelligent agents, reasoning under uncertainty, bio-inspired optimisation, and machine learning.

3. Cloud Computing

Course Title: Cloud Computing
Credit Points: 12.00
Course Coordinator: Dr. Qiang Fu
Course Coordinator Phone: .
Course Coordinator Email: qiang.fu@rmit.edu.au

Course Description
Cloud Computing is a large-scale distributed computing paradigm which has become a driving force for information technology over the past several years. The exponential growth data size in scientific instrumentation/simulation and social media has triggered the wider use of cloud computing services.

This course covers topics and technologies related to Cloud Computing and their practical implementations. You should explore different architectural models of cloud computing, the concepts of virtualisation and cloud orchestration. You should gain hands-on experience with various features of popular cloud platforms such as Amazon Web Service throughout the lectorials, tutorials, and laboratory sessions. Advanced cloud programming paradigms such as Hadoop�s MapReduce is also included in the course. You should also learn the concept of modern Big Data analysis on cloud platforms using various data mining tools and techniques. The lab sessions cover cloud application development and deployment, use of cloud storage, creation and configuration of virtual machines and data analysis on cloud using data mining tools. Different application scenarios from popular domains that leverage the cloud technologies such as remote healthcare and social networks will be explained. The theoretical knowledge, practical sessions and assignments aim to help you to build your skills to develop large-scale industry standard applications using cloud platforms and tools.

This course focuses on learning emerging issues related to Cloud computing technology.�

4. Programming Project 1

Course Title: Programming Project 1
Credit Points: 12.00
Course Coordinator: A/Prof. Vic Ciesielski
Course Coordinator Phone: +61 3 9925 2926
Course Coordinator Email: vic.ciesielski@rmit.edu.au

Course Description
This course is designed to provide you with hands-on practical experience developing software and/or delivering IT solutions in a project environment. You will work in teams and complete projects from inception to implementation. The emphasis is on understanding and working within a corporate environment, using formal project and software delivery methodologies and integrating all the skills and knowledge that you have acquired into a solid base from which to progress your professional life.

This course includes a work integrated learning (WIL) experience in which your knowledge and skills will be applied and assessed in a real or simulated workplace context and where feedback from industry and/ or community is integral to your experience.

5. Programming Project 2

Course Title: Programming Project 2
Credit Points: 12.00
Course Coordinator: Dr Andy Song
Course Coordinator Phone: +61 3 9925 9761
Course Coordinator Email: andy.song@rmit.edu.au

Course Description
This course is designed to provide you with hands on practical experience developing software and/or delivering IT solutions in a project environment. You work in teams and will complete projects from inception to implementation. As the project teams and projects will be agreed during the first week, it will not be possible to transfer to or from this course after week 1.

The emphasis is on understanding and working within a corporate environment, using formal project and software delivery methodologies and integrating all the skills and knowledge that you have acquired into a solid base from which to progress your professional life.

Note 1: Unless you are required to complete a one year project, do not enrol in both COSC2408 and COSC2409 as you usually only need to complete either, not both, of these courses. Check with the School for further clarification.

Note 2: This course includes a work integrated learning experience in which your knowledge and skills will be applied and assessed in a real or simulated workplace context and where feedback from industry and/ or community is integral to your experience.

Electives/ Computer Science Options:

Agent-Oriented Programming and Design

Course Title: Agent-Oriented Programming and Design
Credit Points: 12.00
Course Coordinator: Dhirendra Singh
Course Coordinator Phone: N/A
Course Coordinator Email: dhirendra.singh@rmit.edu.au

Course Description
The course provides a foundation in agent-based modelling and simulation techniques for understanding complex systems. Agent-based models (ABMs) work by representing the entities (agents) of a system and their interactions from which system-level phenomena emerge. Examples of emergent phenomena that can be represented with ABMs include flocking in birds, traffic congestion on roads, and spread of infection in a population.

In this course you will learn how to design, implement, use, analyse, and critique agent-based models of complex systems. You will construct �what-if� scenarios, and implement �policy interventions�, to examine how exogenous changes can perturb the system, and use critical analysis to determine likely impacts of proposed changes.

This is a studio-style course, with a strong emphasis on learning by doing.

Please note that if you take this course for a bachelor honours program, your overall mark in this course will be one of the course marks that will be used to calculate the weighted average mark (WAM) that will determine your award level. This applies to students who commence enrolment in a bachelor honours program from 1 January 2016 onwards. See the WAM information web page for more information.(http://www1.rmit.edu.au/browse;ID=eyj5c0mo77631)

Data Communication and Net-Centric Computing

Course Title: Data Communication and Net-Centric Computing
Credit Points: 12.00
Course Coordinator: Dr Fengling Han
Course Coordinator Phone: +61 3 9925 3402

Course Coordinator Email: fengling.han@rmit.edu.au

Course Description
This course aims to provide a broad introduction to the fundamentals of data communication and network technology. The emphasis is on data communication from the perspective of computer scientists and information technologists. The course covers the underlying mechanisms and their characteristics that need to be considered by communicating application software.

Topics include:

principles and reference models of data communication,
basic operation of communication systems,
protocols, error handling and�applications in networked environments.

Database Systems

Course Title: Database Systems
Credit Points: 12.00
Course Coordinator: Xiuzhen Jenny Zhang
Course Coordinator Phone: +61 3 9925 2774
Course Coordinator Email: xiuzhen.zhang@rmit.edu.au

Course Description
The course aims to deepen your knowledge of database systems. You will cover the design and implementation of file structures, indexing and analytical details that will enable you to understand efficiency in query evaluation. The concept of the transaction is introduced along with the necessity of concurrency control and serialisability. Issues of backup and recovery for databases are also introduced.

Deep Learning

Course Title: Deep Learning
Credit Points: 12.00
Course Coordinator: Ruwan Tennakoon
Course Coordinator Phone: +61 3 9925 3306
Course Coordinator Email: ruwan.tennakoon@rmit.edu.au

Course Description
Deep Learning is a field of Machine Learning that focus on large scale neural networks. Deep Networks are suited for solving a variety of complex problems, such as computer vision, natural language processing, and large-scale state estimation. While Deep Networks require vast

collections of training data, the networks often outperform humans. Deep Learning has increasingly become a core aspect of both major IT companies and new tech start-ups.

You will undertake a thorough study of the Deep Learning, from its foundations in perceptron�s and multi-layer networks, through to present-day deep architectures including convolutional neural networks. You will critically analyse issues with deep learning, learn how to use open source toolkits, and learn how to critically analyse outputs from these applications. Assessed work will involve real world datasets from a variety of domains.�

Peer-to-peer Networks

Course Title: Peer-to-Peer Networks
Credit Points: 12.00
Course Coordinator: Dr. Fengling Han
Course Coordinator Phone: +61 3 9925 3402
Course Coordinator Email: fengling.han@rmit.edu.au

Course Description
Peer-to-Peer networks have provided success applications for sharing resource in recent years. Base on the infrastructure of communications technology, started from sharing computing resources, such as bandwidth, storage and processing power, sharing physical sources has become the new trend in peer to peer applications. This course is about advanced network applications and associate networking technologies. It is designed to help you integrate your knowledge of networking and computing to analyse and design effective platforms for network based advanced applications. Topics covered include fundamentals of computing and networking, peer to peer computing and associate applications. You will develop your critical thinking skills in this course to the point where you are able to systematically analyse peer to peer to computing issues at a high level of theoretical abstraction and at the same time be able to apply this knowledge to real life peer to peer computing system design.