

Course descriptions for the Bachelor of IT - 2010

Core courses

Introduction to Information Technology

Introduction to Information Technology covers a range of fundamental key topics in computers technology including hardware, software, internet, multimedia formats, operating systems and management. In this course we assemble and disassemble a computer and understand how each major hardware component communicates with each other. An introduction of operating systems includes exploration of Ubuntu Linux and other UNIX variations. Networking and the internet is explained through a network tour of a real system. Computer security issues are taught using practical examples; how long it takes to crack a password, sending encrypted messages and testing network speeds. You will learn about audio (e.g. MP3, wav), image (e.g. jpeg, tiff) and video (MP4, mov, avi, MPEG) formats, and how they compress and store information.

Introduction to Programming

In an IT-driven world, programming is an important skill for professionals who intend to work in high-tech or IT related industries. Programming is a skill that has its foundation in problem-solving ability. In this course, you will learn basic programming skills in a visual 3D programming environment. On completion of this course, you should be able to analyse computing problems, design suitable algorithmic solutions, and code those solutions in computer programming languages.

In the first part of the course, you will learn basic programming concepts such as variables, loops, decisions, Boolean logic, arrays, objects, and event-driven programming, by using the visual programming environment Alice. Alice is a drag-and-drop programming environment that will help you understand the relationship between programming statements and the behaviour of the program.

In the second part of the course, you will learn how to apply the same programming concepts using the industry-standard Java programming language. This will require you to write programming code manually, using either NetBeans or Eclipse as development environment.

Computer Organisation

This course aims to give students an introduction to digital concepts and number systems, computer architecture, computer operation and assembly language programming which form the basis for the design and operation any modern computer, embedded system, or mobile device. At the end of this course, students have a thorough understanding of fundamental principles such as data storage, data flow, and program structure that form the basis of modern programming languages.

Students gain practical skills and problem solving skills through activities in class, with a focus on number systems, IEEE754 floating point format, error-detection and correction systems, and 68K assembly programming problems. To do the assignments, students are required to write simple, graphical games by using the EASy68K emulator.

Mathematics for Computing

Mathematics for Computing introduces and studies (with an emphasis on problem solving) many of the fundamental ideas and methods of discrete mathematics that are the tools of the computer scientist. The course demonstrates the importance of discrete mathematics for computer science.

Web Programming

With the growing importance that the Internet has on all of our lives, there is a greater importance for students to understand the nuances involved in the workings of dynamic Web applications. This course introduces students to some of the basic concepts of the World Wide Web and covers some of the protocols needed to support the development of dynamic web applications.

Specifically, the course starts with an overview of Internet technology and will later introduce you to some of the latest Web protocols and standards such as XHTML, client side processing with ECMA Script (JavaScript) and server side processing with PHP. At the end of the course, students are expected to be equipped with some of the skills required to create modern web 2.0 applications such as social networking sites and ecommerce sites.

Programming 1

This course teaches the fundamental concepts of the object-oriented approach to programming through the widely-used Java programming language. The concepts include classes and objects, encapsulation, inheritance, polymorphism, abstract classes and interfaces. Concentrating on aspects of Java that best demonstrate object-oriented principles and good practice, you will gain a solid basis for further study of the Java language, and of object-oriented software development.

This course also continues the topic of algorithm development using standard control structures, and design methods such as step-wise refinement that were introduced in Introduction to Programming. In addition, the course introduces techniques for code reuse, basic strategies for software testing and debugging, exception handling and text file reading and writing.

The assignments in the course are designed around a complete practical application such as a Library System. Each assignment builds on previous assignment(s) until a complete system is developed and put into use.

Database Concepts

This course is an introduction to the principles, use, and applications of database systems. Students who complete the course will be able to design and create databases, be able to extract information from databases, understand in broad terms how database systems work, and understand the purposes for which databases are used.

At the end of this course, students can understand and explain the fundamental concepts, theories and methods of the relational data model. Assignments and lab exercises give students the opportunity to work with different databases hosted on an Oracle server, and gain experience with SQL; leading up to the skills required for making business applications that involve data warehousing, data marts, and data mining.

Software Engineering Fundamentals

This course aims to guide you in both the theoretical and practical aspects of developing computer solutions for real-world problems. You will study common tools and techniques used in current software engineering methodologies. The course is assessed through examination and case

studies, where you get to apply your understanding of software engineering and use the tools and techniques discussed in class. Many students take advantage of the project management techniques taught to increase their management potential in industry.

The course takes a freshman software engineer through the various disciplines involved in creating software. The focus is on the aspects of software engineering methodologies that have been proven to work in industry. You will learn about “chickens and pigs”, “sprints and demos”, along with traditional requirements, analysis and modelling. The goal is for students to be able to adapt to problems and resolve them using the tools from the class.

Data Communication and Net-centric Computing

This course will provide a broad introduction to the fundamentals of data communications and network technology. Emphasis is on higher aspects of data communications from perspective of the computer scientist and information technologist; the course will cover the underlying mechanisms and their characteristics that need to be considered by communicating application software.

Topics include: Internet protocols, physical aspects of data communications, data encoding, error detection, multiplexing, flow and error control, routing, LAN and Medium Access Control (MAC) methods, wireless networking, Wide Area Networks, packet and circuit switching, ATMs, transport protocols, emerging networking technologies, as well as applications in networked environments, network design, and IP addressing schemes.

Programming 2

This course extends the study of basic programming principles introduced in Programming 1. This course extends the study of basic programming principles introduced in Programming 1. More advanced concepts of object-oriented design and implementation in Java will be introduced. On completion of this course you should have gained competence in the use of the Java programming language and the essential parts of the Java class library and be able to develop small to medium sized application programs that demonstrate professionally acceptable coding and performance standards. In addition, you will have knowledge of basic algorithms and data structures, in particular the application of recursive algorithms, simple searching and sorting algorithms and simple linked list structures, and an introductory understanding of event-driven programming and graphical user interfaces.

Professional Computing Practice

This course is an introduction to computing ethics, law and marketing. It is intended for IT students who have not studied business principles, or who have little work experience in the industry. The course provides a survival kit for IT graduates entering the work force. The course considers computer ethical issues, such as information privacy, computer crime, computer terrorism. The course considers the international legal framework available to protect software system development. This includes non-disclosure agreements, employment contracts, intellectual property law (copyright, patent, licensing, and royalties), trademarks and warranty disclaimers. The course also considers the marketing of a software system development, involving SWOT analysis and action plan (including budget).

Software Architecture: Design and Implementation

This course provides a good mix of theory and practice of client-server architecture which helps you build production-ready distributed applications in Java. The first week covers advanced Java

features such as generics and the Java Collection Framework. The main topics of the course include application design and design patterns, multithreaded programming, database programming with JDBC, distributed programming with socket and RMI and introductory JSP/servlet web application development.

Throughout the course, you will be guided to work in groups to develop non-trivial client-server applications such as multiplayer networking games, (video) instant messaging systems and ecommerce web applications. This course will provide the necessary foundation for the more specialized course Ecommerce and Enterprise Systems.

Internship (industry placement)

This course is the Internship course. The aim of the Internship is to gain practical experience in industry and to apply all theoretical and fundamental concepts learned on campus. Students will be supervised intensively by a qualified supervisor and additional help and resources are provided when necessary.

Prior to the internship period, each student is assisted to obtain an internship position at a well-known company in the industry, in which the student will have opportunities to participate in training programs and take part in real projects to gain practical experience, to apply all theoretical and fundamental concepts learned on campus, also to acquire new knowledge and skills.

Many RMIT interns are offered fulltime positions at the internship company when the internship period is over.

Course descriptions for the Application Programming Major

Programming Techniques

This course extends the study of programming principles developed in pre-requisite courses, including the use of defensive programming, debugging, testing, coding standards and practices; this material is presented using the C programming language as a vehicle for instruction. This course provides an excellent basis for learning how to develop iPhone applications.

On completion of this course students will be capable of implementing small to medium programs of varying complexity, using the most commonly used features of the C language; employ good programming style, and standards and practices during program development; organise separate source translation units properly, for larger programs, so that they reflect the use of Abstract Data Types wherever required; use simple, possibly dynamically allocated, data structures in solutions; and adapt programming experience and language knowledge to other programming language contexts.

Programming using C++

This course introduces C++ as an object-oriented programming language. The aim is to help students gain a better understanding of OO design and program implementation by using OO language features. Students learn to understand object-oriented programming features in C++, apply these features to program design and implementation, gain practical experience with C++, and understand the role of patterns in object-oriented design.

Throughout the semester, students acquire unique skills in using a UNIX development environment and the “make” build tool. Students work on tasks directly related to specific object-oriented programming techniques and C++. For the assignments, students are required to design and implement medium-sized applications with a focus on re-usable design, high performance, and problem solving skills. The assignments will include game-related algorithms and other typical problems that are commonly solved with the use of C++.

Electronic Commerce and Enterprise Systems

This course introduces concepts, technologies and patterns necessary to design and implement large-scale electronic commerce and enterprise systems in J2EE. Specifically, you will learn about developing managed business and data access components with EJB technology, building web front-end with JSP and Servlets, and applying enterprise application patterns in designing your applications.

Throughout the course, you will have chance to apply your knowledge in building full-scale distributed web-based applications using J2EE. Previous classes in the course have seen students build a web-based ecommerce storefront, a school management system, and an online dating application.

Web Development Technologies

This course provides you with a strong foundation in building web-based applications running on top of the Microsoft .NET Common Language Runtime. Topics covered include the .NET framework, the C# 1.x/2.0/3.0 programming language, the ASP.NET 3.5 web application framework, the ADO.NET data access library, LINQ, SQL Server 2005 programming, Crystal Reports reporting solutions, and .NET XML web services.

At the end of the course, you will have built a real-world, production-ready ASP.NET web application which involves using the various technologies learned throughout the course. The foundational knowledge taught in this course enable you to quickly catch up with other development platforms on the .NET framework such as WinForms, WPF and Silverlight.

Software Testing

This is an introduction to software testing principles and practices in industry. We will discuss practical software testing goals and approaches to testing software at various levels - from end user application level to source code level through all phases of the Software Development Lifecycle. The course material will include the following - software quality standards and metrics, types of testing, function and performance testing, test planning, test case generation, shipping criteria, estimating test resources, scheduling tests, object-oriented factors in testing, assessing risks, prioritising tests, automation strategy, defect management, test execution, the end game before final shipping and post ship test and support issues. Software quality assurance activities will be discussed as part of a dynamic process that is flexible and constantly tuned to the changing needs of a project.

Course descriptions for the Multimedia Design Major

Design for Interactive Media 1

This course gives an introduction to fundamental concepts that will develop your critical and creative skills for multimedia. You will be introduced to a wide range of fundamental concepts for visual cultural theories including representation, reproduction, metaphor, semiotics, narrative, time, and visual language, involved in creating work for multimedia.

Class time will be divided between critiquing and analysis of visual theory, images, films, interactive media, alongside development of creative content for visual based media, concept development and animation skills useful in the creation and presentation of multimedia.

At the completion of this course you should have developed a competency in digital media authoring, have an ability to focus and develop creative solutions involving digital media authoring, and be competent in visual language and image skills. It is expected you will have gained skills and confidence to challenge and develop your creative and critical ability.

Design for Interactive Media 2

This course builds on fundamental concepts for multimedia and visual literacy introduced in Design for Interactive Media 1. You will be introduced to basic design aesthetic and interactive authoring principles, composition, and media integration. Supervised workshops will involve critiquing all forms of visual media, practical digital and interactive design demonstrations, as well as concept and skill development.

In this course it is expected that you will achieve the following: production, direction, composition, and documentation of multimedia based works; media integration techniques; concept, theories and methodologies of composition; group project work; authoring techniques – including interaction and navigation; intermediate skills in analysis of visual culture and theories of postmodernism.

The focus will be on creative solutions involving digital media authoring, development of visual language and image skills. You will be encouraged to challenge and develop your creative and critical ability through class exercises, individual and group projects.

Time Based Media

The course is designed to introduce you to the fundamental issues involved in using and manipulating time for dramatic effect in multimedia and linear works. It will enable you to develop your understanding of time-based media through topics including: comparisons of temporal and static imagery; the use of time in film; edited vs non-edited time; manipulating the presentation of information using time-based events in interactive works; time-based concepts for Internet publishing; and future directions of time-based media.

This course helps you to develop an ability to analyse, evaluate and reflect upon the usage of time in interactive media, films and Internet sites. You will gain a broad appreciation of the possibilities for using time-based media by examining selected published works which illustrate the diverse ways time can be used to communicate ideas effectively. The subject also encourages you to experiment with cinematic editing techniques, and methods to convey the passage of time that can alter an audience's perception of time, thus effecting pace and creating mood.

Imaging and Animation

This course introduces key topics, principles and techniques for 3D imaging and animation, including rendering, virtual environments and the behaviours of objects. The focus will be on the theory of virtual 3D and of virtual worlds and the use of software applications to create 3D animation, dynamics and motion oriented visual design.

You will study background theory of virtual worlds and 3D aesthetics in lecture mode, followed by tutorials in 3D modelling techniques within a creative production environment that mirrors the industry as closely as possible. The course encourages thoughtful artist exploration of the media, celebrates its possibilities and respects well-developed skills. You will develop and refine creative concepts and apply these in practice to project work and assignments that require innovative and effective responses.

Advanced 3D Imaging and Animation

This course further develops key topics, principles and techniques for 3D imaging and animation, including rendering, morphing, virtual environments and the behaviours of objects presented. The focus will be on the use of software applications to create animation, dynamics and motion oriented visual design.

You will develop advanced capabilities in producing and rendering photo-realistic images, creating original animation, storyboarding, and flowcharting and visual development. You will develop skills in the principles of narrative and timing, soundtrack design, and electronic post production. You will learn to use software applications to create animation, dynamics and motion oriented visual art and design. You will acquire skills in integrating computer generated animation for interactive multimedia, interactive display and business presentation.

Course descriptions of other IT electives

Document Markup Languages

eXtensible Markup Language (XML) is a mark-up language used extensively on the World Wide Web. Apart from being one of the major success stories to come out from the Web in the last few years, the language is now finding increased prevalence in business as a medium for structured data exchange and information retrieval. XML skills are now essential in many computer science and information technology roles, and are a must for anyone interested in the next generation of Internet publishing standards.

This course is an introduction to the principles, use, and application of document markup languages, especially for use on the World Wide Web, focussing on XML. Topics include document markup languages (especially XML); document description languages (such as Document Type Definitions (DTDs and XML Schemas); XML namespaces; document transformation and manipulation (using eXtensible Stylesheet Language Transformations (XSLT) and XML APIs, such as the Simple API for XML (SAX) and the Document Object Model (DOM); and XML-based technologies such as RSS, ATOM, and RDF.

Software Engineering: Process and Tools

This course aims to guide students in both the theoretical and practical aspects of developing computer solutions for problems. This includes understanding working in project groups, project management, producing large programs via a Software Engineering process, testing and metrics. On completion of this course you should be able to explain processes involved in large scale software system engineering, create tests for software components, demonstrate knowledge of project management issues, demonstrate working in a project group, and demonstrate knowledge of how to run and document meetings.

Software Engineering 2

Software Engineering 2 builds on the work of previous software engineering courses by examining software design and software architecture in much greater detail. Throughout the semester, the most important “design patterns” and best known object-oriented design and programming practices, pioneered by the Gang of Four in the 90s, are covered.

Students are taught and apply agile estimation and project management. This means students have stand-up meetings, peer reviews, sprints every two weeks, estimation, product backlogs and workflows.

All of this is tied together with a semester long real life software engineering problem where students must show they have mastered best software engineering principles and practices. In a previous semester, students designed and implemented a graphical Traffic Simulator.

Web Servers and Web Technology

This course introduces you to the role of a web server administrator and the related fundamental concepts. You will study the technical aspects of: networking and the Internet, the HTTP and related protocols, web server performance, load balancing, web server security issues, web caching and proxies, and content negotiation (e.g. content type and language.)

The course strikes an excellent balance between theory and practice by providing students with opportunities to work with the Apache 2.0 web server, in a UNIX environment. You will apply

concepts by fulfilling web server administrator tasks including analysing requirements, installing and configuring a web server, measuring web server performance, and deploying security features.