Program Guide

Bachelor of Information Technology (Games and Graphics Programming)

School:	School of Computer Science and Information Technology
Portfolio:	Science, Engineering, and Technology
RMIT Program Leader:	Dr. Santha Sumanasekara (santha.sumanasekara@rmit.edu.au)
Local Contact Academic Contact:	Jennifer Lade (jennifer.lade@rmit.edu.au)
For more information:	Selection Officers: George Fernandez (gfernandez@rmit.edu.au) and Phil Vines (phil@cs.rmit.edu.au) School administration office: 10.10.07 Telephone: 9925 2348

1. Program title, Program Number and CRICOS Code

BP215 – Bachelor of Information Technology (Games and Graphics Programming)
CRICOS code: 052661K

2. Abbreviated title of award

BInfoTech (GamesGraphicsprog)

3. External Accreditation

This program is accredited at professional level by the Australian Computer Society, which accredits Information and Communication Technology related programs in Australia.

4. Statement of capabilities

The creative industries have seen increasing demand for appropriately qualified graduates. Demand is particularly evident in the electronic games industry where digital artists, designers and programmers, working together, produce material for a variety of platforms including PCs consoles and mobile devices. This program is designed to respond to the needs of the creative and graphics industries, and more specifically focuses on the electronic games industry. Students study in an exciting environment with practicing artists, designers and programmers, and can expect to graduate with both broad and specific skills applicable to many areas.

The Bachelor of Information Technology (Games and Graphics Programming) develops the capabilities required for students wanting to pursue games or graphics programming careers. It does this in the context of a broader computer science and software engineering framework, applicable to the IT industry in general. A key feature of the program is that students undertake projects and core courses with students from Digital Art and Games Graphics Design programs. Access to courses from these two streams, and further courses in Computer Science, are available by means of electives.

This program is jointly offered by the School of Computer Science and Information Technology and the School of Creative Media. Both of these schools have a strong tradition of “hands on” teaching, providing students with the opportunity to mix course content and practical experience. This approach, coupled with our close involvement with industry, produces graduates who are highly regarded in the workplace. The curricula used in our various degree programs reflect these needs incorporating cutting-edge technologies while maintaining a good coverage of theoretical and algorithmic foundations of computer science, information technology, software engineering, multimedia design, and digital arts. The Games and Graphics Programming degree, in particular, develops a skills set that spans from theoretical and algorithmic foundations to

cutting-edge developments in various aspects of Computer Games industry, and graphics programming in general.

This program was developed based on the recommendations of the curriculum development joint task force of IEEE Computer Society and Association of Computing Machinery. It was developed though a formal consultation process with a number of stakeholders, including the School's Industry Advisory Committee, academic staff of the School, alumni, and current students. This process resulted in the identification of the following graduate capabilities that are required of a games and graphics programming graduate to successfully engage in a professional capacity in the relevant field of industry of the 21st century.

The graduate capabilities developed by the Games and Graphics degree program are composed of the following dimensions:

- **Enabling Knowledge**
  This capability allows one to apply knowledge effectively to new situations and learn from the experience.

- **Critical Analysis**
  In general, this capability allows one to examine and consider accurately and objectively any topic, evidence, or situation.
  More specifically, this capability allows one to:
    - Analyse and model requirements and constraints for the purpose of designing and implementing software systems;
    - Evaluate and compare designs of such systems on the basis of requirements of the organisational needs.

- **Problem Solving**
  In general, this capability allows one to analyse problems and synthesise suitable solutions.
  Specifically, this capability allows one to:
    - Design and implement software systems that accommodate specified requirements and constraints, based on analysis or modelling or requirements specification

- **Communication**
  In general, this capability allows one to communicate effectively with a variety of audiences through a range of modes and media.
  Specifically, this capability allows one to:
    - Present and explain complex software systems solutions, alternative solutions, and decision recommendations to both IT and non-IT personnel via technical reports of professional standard and technical presentations.

- **Team Work**
  In general this capability allows one to work as an effective and productive team member in a range of professional and social situations.
  Specifically, this capability allows one to:
    - Work effectively in different roles, to form, manage, and successfully produce outcomes from teams, whose members may have diverse cultural backgrounds and life circumstances, and differing levels of technical expertise.

- **Responsibility**
  In general this capability refers to accepting responsibility for one's own learning and make informed decisions in judging and adopting appropriate behaviour in professional and social situations. This includes accepting the responsibility for life-long learning.
  Specifically, this capability allows one to:
    - Effectively apply relevant standards, ethical considerations, and an understanding of legal and privacy issues to designing software systems.

5. Program structure

The Games and Graphics Programming degree program comprises of following components.

Year	Semester	BP215 Bachelor of Information Technology (Games Graphics Programming)			
1	Sem 1	Games Studio 1	Introduction to Programming	Mathematics for Computing	Web 3D Technologies
	Sem 2	Games Studio 2	Programming Techniques	Mathematics for Advanced Computing	Program Elective
2	Sem 1	Data Communication and Net-centric Computing	Interactive 3D Graphics and Animation	Algorithms and Analysis	Program Elective
	Sem 2	Network Programming	Real-time Rendering and 3D Games Programming	Java for C Programmers	Program Elective
3	Sem 1	Software Engineering Fundamentals	Interactive Digital Media Project A	Student Elective	Program Elective
	Sem 2	Object Oriented Programming	Interactive Digital Media Project B	Digital Media Computing	Student Elective

The core of the Games and Graphics Programming program consists of two major components: computer science core and games and graphics programming-specific courses.

The computer science core courses provide a solid theoretical and algorithmic foundation for all essential computer science topics and cover the knowledge units that every Computer Science student must have studied. This component forms a superset of core units identified in CC2001 ACM curriculum guidelines[1]. The proposed computer science core will be about 120 (10 X 12CP courses) credit points in size.

There are eight Games Programming-specific courses in the program, giving you the opportunity to learn many different aspects of games programming. There are two important milestones in the program: completion of Games Studio courses in the first year and the completion of Interactive Digital Media project in the third year.

A key feature of this program is that you undertake projects and courses with students from Digital Arts and Games and Graphics Design programs. Program Electives will enable you to experience other aspects of the games industry, such as Digital Arts and Games Design. The program structure allows you to take four program electives. In addition to electives from Digital Arts and Games Design, you will be able to choose some other computer science electives as well.

In addition to these, the University policies and regulations (http://mams.rmit.edu.au/i9izc93yhl0tz.pdf) require that you should do a minimum of two student electives (free-choice electives) available from any discipline (availability of electives to be published by the University at the beginning of each year). There will be a provision of two student electives in the proposed program structure. However, it is noted that you may, if you wish, choose computer science electives in place of student electives.

Program Electives:

COMM2301	Media Cultures 1
COSC1009	Design for Interactive Media 1
GRAP2169	Imaging & Animation
COMM2302	Media Cultures 2
GRAP2326	3D Char: Create & Animate & FX

[1] http://www.computer.org/education/cc2001/final/index.htm

COMM2305	Sound Design
ISYS1057	Database Concepts
COSC2413	Web Programming
COSC1093	Scripting Language Programming
COSC1011	Design for Interactive Media 2
COMM2244	Narrative and Communication
COSC2276	Web Development Technologies
COSC1114	Operating Systems Principles
COSC1301	Web Servers and Web Technology
COSC1301	Web Servers and Web Technology
COSC2309	Mobile Application Development
COSC2413	Web Programming
COSC2391	Programming 3
COSC2276	Web Development Technologies
VART2963	Digital Curation & Exhibition Practice
VART2964	Digital Painting, Light & Print
COSC1093	Scripting Language Programming
COSC1114	Operating Systems Principles

Program Capabilities Map:

All undergraduate programs within the School of CS&IT aim to develop the following capabilities, to varying degrees, in graduates of the programs:

Enabling Knowledge, Critical Analysis, Problem Solving, Communication, Team Work, and Responsibility.

Enabling Knowledge: **You will acquire all of the capabilities below using a foundation of relevant concepts and knowledge. This enabling knowledge is communicated in the program's courses as follows:**		
Courses in which key abilities or skills that contribute to the development of this dimension of capability are taught, have learning activities and are assessed	The specific aspect of the capability developed in each course	Semester in which this is offered
Programming 1	Syntax and basic features of the object-oriented programming language Java; good programming	1

	style, standards and practices in programming; the use of standard Java classes, interfaces, containers; and basic techniques for code reuse and testing.	
Computer Organisation	structure and function of digital computers, operating systems and basic assembly language programming.	1
Mathematics for Computing	Key areas of discrete mathematics and their underlying application in computing.	1
Database Concepts	Fundamental database concepts including analysing, designing, defining, constructing and manipulating relational database systems.	1
Programming 2	Advanced concepts of program design, professionally acceptable coding and performance standards, fundamental algorithms and data structures, event-driven programming and graphical user interfaces.	2
Web Programming	the client-server internet model, W3C standards, mark-up languages, client-side scripting, server-side scripting to access and manipulate data, database access and internet security.	2
Software Engineering Fundamentals	An object-oriented software engineering process, and how to apply it to design a specified software system Experience with industry-standard methodologies and tools for analysing and modelling software system requirements Practice in following industry-standard templates for documenting the design and implementation processes Teamwork principles, team roles and dynamics, project planning techniques Theoretical and practical knowledge of the software engineering process, models and processes for design of software systems, methods of evaluation; and current trends in the areas of software engineering.	2
Data Communication and Net-centric Computing	higher level aspects of data communications and network technology, details of the underlying mechanisms, principles, reference models, protocols, error handling, appropriate for the development of software applications in a networked environment.	2
Programming Techniques	experience with modular programming using the C programming language; good programming style, standards and practices, and the implementation of dynamic data structures in a modular fashion in C.	3
Software Engineering: Process and Tools	software engineering processes required for the development of large-scale software systems; techniques for software testing and metrics, team work, managing teams, group dynamics, leadership and responsibility.	3
Computing Theory	Foundational issues in computer science, computability, time requirements for computation, probabilistic approaches to difficult problems, principles of cryptography, language grammars, and models of computation.	3
Algorithms and Analysis	the operation, implementation and performance of fundamental algorithms and data structures, and the relative merits and suitability of each for various	4

	applications.	
Operating Systems Principles	Fundamental principles, strategies and algorithms used in the design and implementation of operating systems.	4
Professional Computing Practice	Basic concepts of business organization and business practices; ethical, legal and marketing issues relevant to the IT industry.	4
Database Systems	database query optimisation, transactions, failure and recovery, concurrency control, physical database design, and information retrieval.	5
Artificial Intelligence	artificial intelligence techniques including search heuristics, knowledge representation, planning and reasoning.	5

Graduate Capability: Critical Analysis:		
Courses in which key abilities or skills that contribute to the development of this dimension of capability are taught, have learning activities and are assessed	The specific aspect of the capability developed in each course	Semester in which this is offered
Programming 1	Ability to analyse and model requirements for solving algorithmic computing problems.	1
Database Concepts	Ability to analyse data modelling problem specifications and derive alternative conceptual models that represent the problem in different perspectives leading to alternative database designs.	1
Computer Organisation	Ability to analyse and model the structure and functioning of a digital computer, including overall system architecture, operating system, and digital components.	1
Mathematics for Computing	Development of analytical and critical thinking abilities suitable for mathematical and computing problems.	1
Programming 2	Ability to analyse requirements and constraints for the design of small-scale software systems.	2
Web Programming	Ability to analyse and model requirements and constraints for the design of client-server internet applications.	2
Software Engineering Fundamentals	Ability to identify, analyse and evaluate requirements and constraints for the design of software systems.	2
Data Communication and Net-centric Computing	Ability to analyse communications systems that use the TCP/IP protocol suite and the abstract 7-layer OSI reference model.	2
Programming Techniques	Ability to analyse and model requirements and constraints for the design of modular software systems.	3
Software Engineering: Process and Tools	Ability to identify and analyse (as part of a team) system requirements for large-scale software systems.	3
Computing Theory	Ability to analyse and compare the characteristics of different types of computational problems and of different models of computation.	3
Algorithms and Analysis	Ability to analyse data structures and algorithms, to compare and evaluate them with respect to time and space requirements, in order to make the most	4

	appropriate design choices for various application areas.	
Operating Systems Principles	Ability to analyse and evaluate operating system functions	4
Professional Computing Practice	Ability to evaluate computing practice case studies and to prepare SWOT analyses.	4
Database Systems	Ability to analyse database queries and data specifications to design appropriate query optimisation mechanisms, and to methodically explore issues of efficient designs for various database systems.	5
Artificial Intelligence	Ability to analyse search and planning problem specifications for the design of solutions that incorporate AI techniques.	5

Graduate Capability: Problem Solving:		
Courses in which key abilities or skills that contribute to the development of this dimension of capability are taught, have learning activities and are assessed	The specific aspect of the capability developed in each course	Semester in which this is offered
Programming 1	Ability to design and implement computer programs to solve algorithmic computing problems, based on analysis and modelling of requirements.	1
Database Concepts	Ability to design and implement database solutions for various application areas and to build queries for users' needs, based on analysis of data modelling problem specifications.	1
Mathematics for Computing	Ability to solve practical mathematical problems.	1
Programming 2	Ability to design and implement small-scale software systems, based on analysis of requirements and constraints.	2
Web Programming	Ability to design and implement a client-server internet application that accommodates specific requirements and constraints, based on analysis, modelling or requirements specification.	2
Software Engineering Fundamentals	Ability to design software systems that accommodate specific requirements and constraints, based on analysis, modelling or requirements specification.	2
Data Communication and Net-centric Computing	Ability to apply knowledge of data communications concepts to solve problems involving data communications flow control and error control methods, and to apply mathematical/analytic skills to basic performance evaluation, utilization, throughput and delay.	2
Programming Techniques	Ability to design and implement modular software systems, based on analysis of requirements and constraints.	3
Software Engineering: Process and Tools	Ability to design and implement (as part of a team) large-scale software systems, based on identification and analysis of requirements.	3
Computing Theory	Ability to design and implement solutions to a wide range of problems, including constructing grammars or automatons for given formal languages and specifying formal languages given grammars or automatons.	3
Algorithms and Analysis	Ability to design and implement efficient software	4

	solutions for various application areas using appropriately selected algorithms and data structures.	
Operating Systems Principles	Ability to design and implement software that employs synchronised concurrent processes	4
Database Systems	Ability to design and implement database storage and index structures,	5
Artificial Intelligence	Ability to design and implement appropriate solutions for search problems and for planning problems (such as determining a sequence of actions for a robot), incorporating suitable AI techniques.	5

Graduate Capability: Communication		
Courses in which key abilities or skills that contribute to the development of this dimension of capability are taught, have learning activities and are assessed	The specific aspect of the capability developed in each course	Semester in which this is offered
Programming 1	Ability to explain key concepts of object-oriented programming in Java, standard classes and interfaces, code reuse and strategies for software testing, in written form, to IT specialists.	1
Computer Organisation	Ability to explain the structure and functioning of a digital computer system, operating system and various digital components, in written form, to IT specialists.	1
Mathematics for Computing	Ability to communicate in written form your knowledge of discrete mathematics, probability and statistics.	1
Programming 2	Ability to explain advanced concepts in program design, coding and performance standards, standard algorithms and data structures, event-driven programming and user interface design, in written form, to IT specialists.	2
Web Programming	Ability to motivate and explain internet application concepts, relevant alternatives and decision recommendations, in written form, to IT specialists.	2
Software Engineering Fundamentals	Ability to explain key ideas and processes in software engineering, in written form, to IT specialists.	2
Programming Techniques	Ability to explain modular programming concepts, relevant alternatives and decision recommendations, in written form, to IT specialists	3
Software Engineering: Process and Tools	Ability to conduct project group meetings and run document meetings.	3
Computing Theory	Ability to discuss computational problems and models of computation, in both written form and classroom discussions, and to write a technical report justifying a course of action from analysis of a scenario.	3
Algorithms and Analysis	Ability to motivate and explain efficient programming concepts, relevant alternatives and decision recommendations, in written form, to IT specialists	4
Professional Computing Practice	Ability to discuss and analyse computing practice scenarios, in both written form and oral presentations.	4

Graduate Capability: Team Work		
Courses in which key abilities or skills that contribute to the development of this dimension of capability are taught, have learning activities and are assessed	The specific aspect of the capability developed in each course	Semester in which this is offered
Programming 2	Awareness of the reality of project team software development as opposed to individual developer effort.	2
Software Engineering Fundamentals	Ability to work as an effective and productive team member in projects that design software systems	2
Programming Techniques	Awareness of the reality of project team software development as opposed to individual developer effort	3
Software Engineering: Process and Tools	Ability to work as an effective and productive team member in projects that develop a large-scale software system .	3
Computing Theory	Ability to work in a small group of peers to solve computational problems	3
Professional Computing Practice	Ability to work in a small group of peers to analyse computing practice scenarios, and to assess the work of peers.	4

Graduate Capability: Responsibility:		
Courses in which key abilities or skills that contribute to the development of this dimension of capability are taught, have learning activities and are assessed	The specific aspect of the capability developed in each course	Semester in which this is offered
Programming 1	Ability to apply relevant standards and ethical considerations to writing computer programs. Developing an awareness of the role and responsibility the individual has with regard to their own learning.	1
Programming 2	Ability to apply relevant standards and ethical considerations to the design and implementation of small-scale software systems. Further development of the imperative for self-directed learning and skills development.	2
Web Programming	Ability to apply relevant standards, ethical considerations, and an understanding of legal issues to designing client-server internet applications.	2
Software Engineering Fundamentals	Ability to apply relevant standards and ethical considerations to designing software systems, especially an understanding of the roles and responsibilities that individuals have within the context of the software project team, and the responsibilities and obligations project teams have to clients.	2
Programming Techniques	Ability to apply relevant standards and ethical considerations to the design and implementation of modular software systems.	3
Software Engineering: Process and Tools	Ability to apply relevant standards and ethical considerations to the design and implementation of large-scale software systems. Further developing understanding of the roles and responsibilities	3

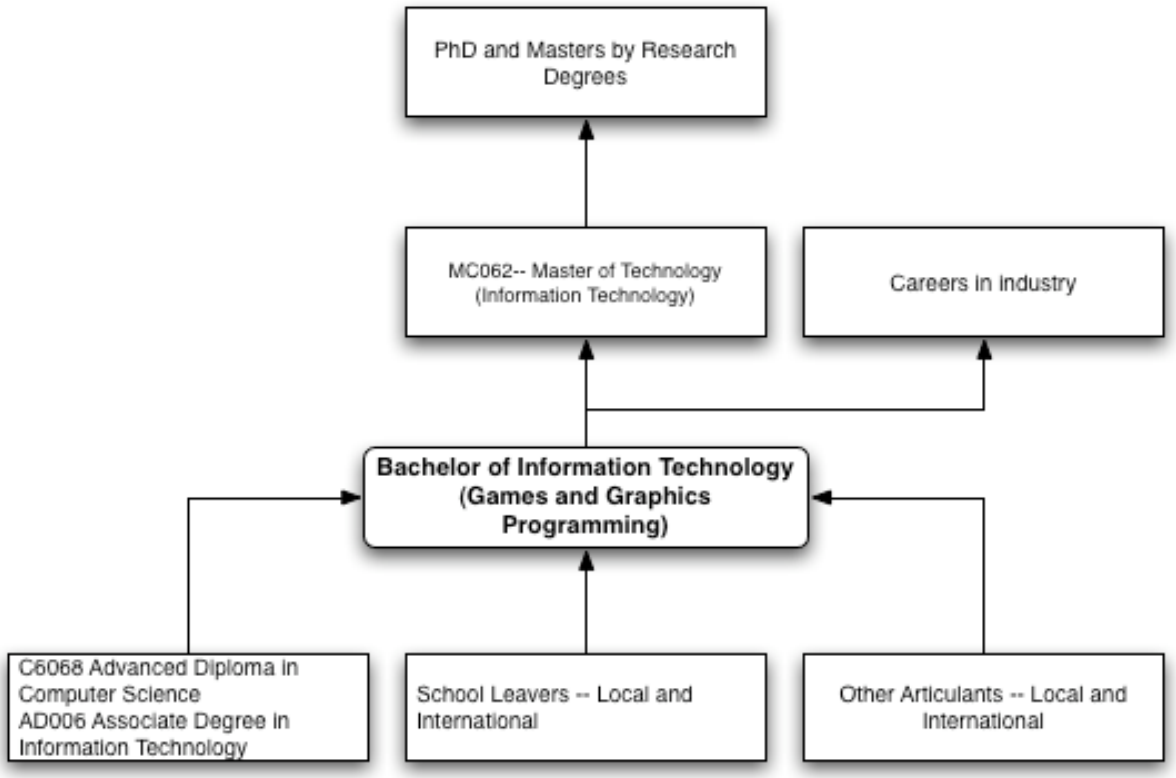
	individuals have within the context of the software project team, and also the responsibilities and obligations project teams have to clients.	
Computing Theory	Developing the understanding of the responsibilities one has in the context of group work and peer assessment.	3
Algorithms and Analysis	Ability to apply relevant standards and ethical considerations to the design and implementation of efficient software solutions.	4
Professional Computing Practice	Ability to apply relevant standards, ethical and social considerations, and an understanding of legal issues to the analysis of computing practice scenarios. Further developing the imperative for the role of self-directed learning and more specifically the role of life-long learning for the longevity and progression of ones career in IT.	4
Database Systems	Ability to apply relevant standards and ethical considerations to the design and implementation of efficient database storage and index structures.	5

6. **Entrance requirements**

VCE Units 3 & 4 English (any) and mathematical methods or specialist mathematics.

Students who obtained study scores above 20 for English will earn selection credit.

7. Program pathways



The following tables indicate how this program articulates with other programs (TAFE, HE – UG and PG) at RMIT highlighting educational pathways, and indicates to students with existing qualifications the extent of exemptions.

Source Program	Owning school	Credit towards this program		Academic requirement for entry	Terms of entry (guaranteed place, merit, etc)	Date of agreement & expiry
		Courses	Time			
C6068 - Advanced Diploma of Computer Science	School of Life and Physical Sciences	- Programming 1 - Programming 2 - Computer Organisation - Mathematics for Computing - Data Communication and Net-centric Computing - Database Concepts - Two Student Electives	1 year	Must obtain a CGPA of 2.0 (Credit Average) or above.	Merit	
AD006 – Associate Degree in Information Technology	School of Life and Physical Sciences	- Programming 1 - Programming 2 - Computer Organisation - Mathematics for Computing - Programming Techniques - Software Engineering Fundamentals - Data Communication and Net-centric Computing - Database Concepts - CCNA Intro - CCNA ICND - Two Student Electives	1Year	Pass	Merit	

Program Destination	Owning school	Credit from this program towards destination program		Academic requirement for entry	Terms of entry (guaranteed place, merit, etc)	Date of agreement & expiry
		Courses	Time			
MC062 – Master of Technology	School of Computer	Programming Techniques	0.5 Year		Merit	

(Information technology)	Science & IT	Software Engineering Fundamentals Algorithms and Analysis One Elective	s			
BH013 – Bachelor of Applied Science (Computer Science) (Honours)	School of Computer Science & IT	No exemptions	0		Meri t	

8. Length of program for FT and PT students

This program will be offered at Melbourne City Campus, where we run two semesters per year. This program is three years full time (24 12-credit point courses). Part-time students should complete equivalent amount of studies, to be completed within 6 years.

9. Program Progression Rules

Detailed Program Structure and Prerequisites.

Year 1

Course Code	Course Title	Credit Points	Pre/co requisites
Core Units – You must complete all of the following courses			
COSC2348	Games Studio 1	12	None
INTE1113	Web 3D Technologies	12	None
COSC1519	Introduction to Programming	12	Unix Survival Skills
MATH1074	Mathematics for Computing	12	None
COSC2349	Games Studio 2	12	COSC2348, COSC1519
MATH2041	Mathematics for Advanced Computing	12	MATH1074
COSC1284	Programming Techniques †	12	COSC1519
First Year Electives -- You must complete one (01) First Year Elective from the list below.			
COMM2301	Media Cultures 1	12	None
COSC1009	Design for Interactive Media 1	12	None
GRAP2169	Imaging & Animation	12	None

Year 2

Course Code	Course Title	Credit Points	Pre/co requisites
Core Units – You must complete all of the following courses			
COSC1111	Data Communication & Net-centric Computing	12	COSC1519
COSC1187	Interactive 3D Graphics & Animation	12	COSC1284
COSC2123	Algorithms and Analysis †	12	COSC1283
COSC1179	Network Programming	12	COSC1284, COSC1111
COSC1226	Real-time Rendering & 3D Games Programming	12	COSC1187
COSC1290	Java for C programmers	12	COSC1284
Second Year Electives -- You must complete two (02) Second Year Electives from the list below.			
COMM2302	Media Cultures 2	12	COSC2348, COSC2349
GRAP2326	3D Char: Create & Animate & FX	12	GRAP2169
COMM2305	Sound Design	12	
ISYS1057	Database Concepts	12	Unix Survival Skills
COSC2413	Web Programming	12	Some programming experience
COSC1093	Scripting Language Programming	12	COSC1284
COSC1011	Design for Interactive Media 2	12	COSC1009
COMM2244	Narrative and Communication	12	
COSC2276	Web Development Technologies	12	COSC1290, ISYS1057
COSC1114	Operating Systems Principles	12	COSC1284
COSC1301	Web Servers and Web Technology	12	Unix Survival Skills

Year 3

Course Code	Course Title	Credit Points	Pre/co requisites
Core Units – You must complete all of the following courses			
COSC2271	Digital Media Computing	12	COSC1290, COSC2413

COSC2350	Interactive Digital Media Project A	12	COSC2349
ISYS1118	Software Engineering Fundamentals	12	Some exposure to programming
COSC2351	Interactive Digital Media Project B	12	COSC2350
COSC1254	Object Oriented Programming	12	COSC1284
Third Year Electives -- You must complete one (01) Third Year Elective from the list below.			
COSC1301	Web Servers and Web Technology	12	Unix Survival Skills
COSC2309	Mobile Application Development	12	COSC1290
COSC2413	Web Programming	12	COSC1290
COSC2391	Programming 3	12	COSC1290
COSC2276	Web Development Technologies	12	COSC1290, ISYS1057
VART2963	Digital Curation & Exhibition Practice	12	
GRAP2326	3D Char: Create & Animate & FX	12	
VART2964	Digital Painting, Light & Print	12	
COSC1093	Scripting Language Programming	12	
COSC1114	Operating Systems Principles	12	
Student Electives -- You must complete two (02) Student Electives			
	Student Elective	12	
	Student Elective	12	

10. Teaching and learning methods

RMIT has a commitment to the principle of student-centred learning: that learning is most meaningful when topics are relevant to your life, needs, and interests and when learning activities actively engage you in creating, understanding, and connecting to knowledge[2].The teaching and learning methods used in this program aim to implement student-centred learning by recognizing that your perceptions of the world are important and relevant, and encouraging you to actively participate in your learning, and to develop solutions in collaboration with your peers. Learning activities include practical exercises, case study analysis, oral presentations, technical and business reports, and individual and group project work.

Lectures (some presented by industry experts) are used to convey some of the basic information necessary for each part of the various courses. Smaller tutorials or laboratory sessions are then used to explore the ideas raised in the lectures, or to give you hands-on experience of technologies. In tutorials, you will often work in a smaller group of about 5 students, to ensure there is real scope for genuinely interactive discussions. Most courses use carefully constructed tutorial questions to illustrate key concepts and to help you develop your understanding.

[2] McCombs, B. and Whistler, J.S. (1997). *The Learner-Centered Classroom and School: Strategies for Increasing Student Motivation and Achievement*. San Francisco:Josey-Bass Publishers.

Course materials (printed course notes, textbooks and reference books) are available from the RMIT Bookshop; the RMIT Library has copies of the books and also provides online access to electronic books and journals; course web pages contain links that let you download worksheets and assignment specifications, email teaching staff, and access message forums, as well as links to external course-related web sites. Lecturers provide additional suitably formatted electronic files and handouts to visually impaired students upon request.

11. Assessment

The school views teaching and learning as a cyclic activity, with assessment and evaluation driving planning and teaching. Assessment is an integral part of learning: information derived from assessment activities is used to facilitate student learning and development, and to improve the quality of the school's programs, services and facilities. Assessment activities examine processes as well as products, and are designed to measure your work against standards, not against other students. As no one assessment can capture the full range of student learning and academic growth, courses use multiple assessments to evaluate what you know and are able to do and to inform adjustments to learning activities.

Assessment is developmental and continuous: that is, you have the opportunity to learn by building on what you already know and are able to do and to carry forward these skills and knowledge to expanded and more complex uses. To reflect industry practice in this area, as you progress through assessments at each level of the program, you are expected to demonstrate at increasingly higher levels of complexity and integration, the knowledge and capabilities set forth in the program standards.

Formative assessment progresses from tutorial exercises and self-test quizzes in foundation courses to participation in seminar discussions, moderated by the lecturer, in some elective courses, to seminar-style discussions in key courses, where you present additional topics in the course material, and apply your knowledge of earlier topics to recognize underlying principles and potential applications of new topics. Some courses involve group meetings and discussions relating to assignments, and participation in case study sessions within groups and with key input and guidance from lecturers and industry experts. Tutorial exercises allow you to explore team dynamics, diagnostics, and management issues.

Summative assessment also becomes more demanding as you progress from foundation courses to electives and key courses:

- The core courses focus on key concepts and initial capability attainment: most assessment activities are based on individual skills and capabilities, and ask you to apply fixed "toolsets" to familiar, well-defined problems, to demonstrate that you have grasped the necessary technical foundations and relevant technologies;
- Elective courses require more complex, open problem-solving, with assignments that require you to design or evaluate solutions for problems with complex or conflicting requirements, or to compare alternative solutions for such problems. In most elective courses, assessment activities also emphasize additional graduate capabilities such as written communication, where you demonstrate that you can integrate concepts and arguments into technical or business reports, or literature reviews of relevant standards, ethical considerations, and applicable research. Some elective courses involve group-focussed assessment.
- Assessment of minor courses vary widely among different minor study streams mainly due to substantial differences in the disciplines covered in these minor study areas. You will be advised by an academic advisor about the specific instruments of assessment once you have chosen the minor study area.

In order to be a lifelong learner, you must be able to evaluate your own work. To support this, some group work is peer-assessed, i.e., following criteria specified by the lecturer, or agreed upon by your class, you assess, and are assessed by, the other members of your group. This is in keeping with student-centred learning, and also helps to alleviate a major misgiving about group work - the possibility of some group members being "carried" by the other members.

Most courses in this program also require you to sit a written examination at the end of the semester, worth between 35% (key) and 60% (foundation) of your final result.

A **portfolio** is a collection of evidence that you prepare to demonstrate mastery, comprehension, application, and synthesis of this program’s concepts. Many of the learning and assessment activities described in this and the previous section can contribute to your portfolio of evidence, in particular, your individual assignments, and your [journal of your] contributions to group activities (case study analyses, presentations, technical and business reports, and group project work).

12. Research, scholarship

RMIT Computer Science & IT has internationally recognised strengths in both applied and theoretical research. Research in the School is focussed almost entirely on software. Most research work is conducted within the five Disciplines:

- The **Data Engineering Discipline** undertakes research into the large-scale storage and access of information, in particular web pages, documents, and multimedia. Research topics include compression, algorithms for information retrieval, genomic information retrieval, indexing of high dimensional data, retrieval for large-alphabet languages, fundamental data structures and algorithms, and pattern matching.

- The **Distributed Systems and Networking Discipline** focuses on providing application, system and networking solutions. Its members conduct research in distributed applications (such as web services, and mobile services), distributed systems (including web servers, middleware, mobile protocols, performance, and security), networking (among others TCP/IP congestion control, mobile networks, and Quality of Service – QoS routing), and graphics (including interactive entertainment, and games).

- The **Intelligent Systems Discipline** focuses on two main areas of intelligent agents and evolutionary computing, with a smaller third area being computational logic. Research topics include agent learning, planning, complex reasoning, agent oriented software engineering, open systems, logics, evolutionary multi-objective optimisation, evolutionary algorithms, data and web mining, and neural and evolutionary approaches to object recognition.

- The **Software Engineering and Programming Discipline** includes research and development work on re-use of software artefacts, improving the elicitation of requirements, measurement of factors that influence software quality, studies of the various aspects of software architecture (for example, relationships, interfaces, and responsibility), improved techniques for software specification, and research into factors that improve outcomes for tertiary computer science and software engineering education.

- The **Web Discipline** undertakes research into technologies supporting the Web and for building web-related applications including document lifecycle management (especially for large websites); web security and internet forensics; engineering web services based on standards such as SOAP and UDDI; improving accessibility, usability, semantics and retrieval; digital media technology (such as watermarking and image/video retrieval); mobile devices; and online education technologies and methodologies.

Much of the research in the above Disciplines will feed directly into courses in the major study streams of this program. For example the Security major will include some advanced courses covering current research topics in computer and internet security and forensics.

The School also has a strong history of undertaking research in teaching practice. Such research in our School has ranged from: exploring the use of new teaching methods for individual courses; the design, development and evaluation of new software tools and instructional aids; web-based tutoring, quizzing and testing systems; academic integrity practices; and online course delivery.

13. Library, IT and specialist resources

RMIT Library already holds or has ordered all prescribed and recommended books. A limited number of copies of books will be available from the Library; some may be available electronically via Safari Bookshelf or electronic journals. This program will be delivered only in on-campus mode.

No additional IT or specialist resources are required to support the new courses. You will use IT facilities within the school. Special software required, such as Rational Rose, is already licensed and installed.

14. Industry links

The School has a very strong Industry Advisory Committee (in some Schools called the “program advisory committee”), which is the main link to industry. Membership includes staff from major IT

companies with global and local presence. The committee meets quarterly and provides feedback on the currency of our programs, the changing needs of Industry and has input into the design of new programs. It also actively contributes to the School through participation in seminars, marketing events, industry awards and scholarships (over $100,000 in Industry Scholarships and awards in 2005).

The School, and in particular the Marketing and Industry Liaison group, works hard to build industry linkages. Examples of our current linkages include:

- *Industry led seminars*. Included in week 0 of Semester 2 is a number of industry led seminars featuring industry contacts including the ACS and alumni discussing issues such as career planning, resume writing, industry trends, entrepreneurship, Intellectual Property, Project management and IT customer service.
- *Industry involvement through sessional teaching***.** The School actively recruits industry expertise for specialist teaching, guest lectures and tutoring. Alumni of the School regularly continue to tutor and teach long after graduation, which adds practical advice and knowledge to the student experience.
- *Professional Associations***.** The Australian Computer Society have strong ties to CS&IT with a senior member on our Industry Advisory Committee and active participation in the Scholarship program (between 4 and 7 ACS/Computer Science Scholarships are awarded to Undergraduate students every year)

15. Support for students' success

A program coordinator will manage the day-to-day running of the program with administrative support from the School of CSIT, and will report to the Undergraduate Program Leader, who is a member of the School's Teaching Committee, Executive Committee, and the Industry Advisory Committee.

The School employs a team of Teaching and Learning Advisors (TLAs) who coordinate and conduct a series of learning support activities for students at all stages of their program. These include:
- Orientation program: Orientation program running in weeks 0 and 1 introduces new students to the specific learning requirements of CS&IT. The program includes workshops and activities to provide students with skills to manage the transition to University learning, an introduction to using the School's computing resources, and opportunities to meet staff and other students.
- Individual support to students to assist you in successful progression through your program. Students identified as 'at risk' work with TLAs throughout the semester for academic and/or personal assistance, in conjunction with other University services. A mentoring program is also run to help improve the programming skills of chosen student cohorts.
- Additional study skills workshop program run during the semester to extend the areas covered in orientation workshops. Workshops are tailored to the specific needs of CS&IT students, and cover both generic study skills and skills specific to courses or student cohorts. Where a particular skill or capability is required for a course, sessions are run within the course timetable.

A Student Progress Committee (SPC) will be scheduled as appropriate to involve all staff teaching into the program. In addition, a program review meeting involving all staff teaching into the program will take place once a year to support collaborative planning and quality improvement.

All undergraduate students in the school have access to the school's computer labs as late as till 9:30PM weekdays. You will be provided with computer accounts on both unix and windows servers which provides an integrated computing environment.

16. Program evaluation and feedback

A staff-student consultative committee (SSCC) will be established and meet three times during each semester to receive comments and feedback from students. Responses and actions will be recorded in the SSCC minutes, which will be available from the School Web site.

17. Student expenses and charges in addition to fees

For on-shore students:
None, apart from standard course fees and university charges.

For off-shore students: Students should seek advice from the local provider.

18. Program Transition Plan

The students in old BP215 Bachelor of Design (Games Graphics Programming) program will be automatically transferred to the new program at the beginning of 2008, and will be graduated with the associated award title. In addition, the following transition rules will apply to students who were on this program prior to 2007 (i.e. those who have started before 2007 structural changes).

1. **All Credit Points earned continue to count towards degree**

Every student will retain credit in these programs for all Credit Points earned and taken pursuant to any previously approved Program Structure.

2. Completion of Core Courses in the old program structure.

Some of the courses have been renamed to better reflect the course content. A list of courses are given below. You are not allowed to re-enrol in a course, if you have already completed the equivalent course.

Code	New Course	Code	Old Course
COSC1073	Programming 1	COSC1073	Programming Principles 1A
COSC1076	Programming 2	COSC1076	Programming Principles 1B
COSC1082	Computer Organisation	COSC1082	Computer Systems 1
ISYS1057	Database Concepts	ISYS1057	Introduction to Database Systems
COSC2413	Web Programming	ISYS1059	Introduction to Internet Technology
MATH1074	Mathematics for Computing	MATH1074	Mathematics for Computer Science
ISYS1118	Software Engineering Fundamentals	COSC1120	Software Engineering 1A
COSC2299	Software Engineering: Process and Tools	COSC2299	Software Engineering 1B
COSC1284	Programming Techniques	COSC1098	Programming Principles 2A
COSC2123	Algorithms and Analysis	COSC1100	Programming Principles 2B
COSC1114	Operating Systems Principles	COSC1114	Operating Systems
COSC1147	Professional Computing Practice	COSC1147	Professional Computing Ethics, Law, and Marketing
COSC2406	Database Systems	ISYS1067	File Structures and Database Systems
COSC1127	Artificial Intelligence	COSC1127	Introduction to Artificial Intelligence

3. Upper Limit of 24 Credit Points for Student Electives ("Free Electives")

The new Structure shows two Student Electives in Year 3.

Students who have taken and passed, or who have been exempted in, two Student Electives in Year 1 and 2 are not allowed to take any more Student Electives.

4. Credit Point mismatches and total Credit Points for Graduation

The normal total of Credit Points for graduation is 288. Students may be left with Credit Point totals which are not exact multiples of 12, due to having taken pre-2006 courses whose Credit Points were not multiples of 12. In such cases, students will not be required to pass more than 288 Credit Points, and may be permitted to graduate with a minimum of 278 Credit Points.

19. Course descriptions

The course guides for the program indicate clearly to students how each course contributes to the overall development and assessment of the program capabilities and how the assessments are related to the development of various capabilities, or the development of specific skills and the acquisition and application of knowledge. Part A of the Course Guide remains relatively static as it establishes the capabilities and learning outcomes from the course. However, the teaching schedule can vary every time a course is offered. The web locators (URLs) of the Part A course guides offered in this program are given in the following table.

Course code	Course name	Part A Course Guide
COSC1204	Agent-oriented Programming and Design	http://www.rmit.edu.au/courses/014061
COSC2269	AI Tools and Applications	http://www.rmit.edu.au/courses/035214
COSC2123	Algorithms and Analysis	http://www.rmit.edu.au/courses/004302
COSC1127	Artificial Intelligence	http://www.rmit.edu.au/courses/004123
COSC1235	Broadcast Networks and Applications	http://www.rmit.edu.au/courses/004232
COSC1082	Computer Organisation	http://www.rmit.edu.au/courses/004075
COSC1107	Computing Theory	http://www.rmit.edu.au/courses/004108
COSC1111	Data Communications and Net-Centric Computing	http://www.rmit.edu.au/courses/004110
COSC2404	Database Administration	http://www.rmit.edu.au/courses/039982
ISYS1057	Database Concepts	http://www.rmit.edu.au/courses/004083
COSC2406	Database Systems	http://www.rmit.edu.au/courses/039983
COSC2271	Digital Media Computing	http://www.rmit.edu.au/courses/035215
COSC2104	Document Markup Languages	http://www.rmit.edu.au/courses/029566
COSC2353	Electronic Commerce & Enterprise Systems	http://www.rmit.edu.au/courses/014053
COSC1207	Evolutionary Computing	http://www.rmit.edu.au/courses/014062
COSC2348	Games Studio 1	http://www.rmit.edu.au/courses/037015
COSC2349	Games Studio 2	http://www.rmit.edu.au/courses/037016
COSC1187	Interactive 3D Graphics & Animation	http://www.rmit.edu.au/courses/004201
COSC2350	Interactive Digital Media Project A	http://www.rmit.edu.au/courses/037024
COSC2351	Interactive Digital Media Project B	http://www.rmit.edu.au/courses/037025
COSC1197	Intro to Distributed Systems	http://www.rmit.edu.au/courses/004207
INTE2425	Introduction to Network Security	http://www.rmit.edu.au/courses/039984
COSC1208	Mathematical Logic and Logic Programming	http://www.rmit.edu.au/courses/014063
MATH2041	Mathematics for Advanced Computing	http://www.rmit.edu.au/courses/028473
MATH1074	Mathematics for Computing	http://www.rmit.edu.au/courses/008418
COSC2309	Mobile Application Development	http://www.rmit.edu.au/courses/036687
COSC1179	Network Programming	http://www.rmit.edu.au/courses/004197
COSC1254	Object-Oriented Programming	http://www.rmit.edu.au/courses/004244
COSC1114	Operating Systems Principles	http://www.rmit.edu.au/courses/004111
COSC1147	Professional Computing Practice	http://www.rmit.edu.au/courses/014058
COSC1073	Programming 1	http://www.rmit.edu.au/courses/004065
COSC1076	Programming 2	http://www.rmit.edu.au/courses/004068
COSC2391	Programming 3	http://www.rmit.edu.au/courses/014052
COSC2408	Programming Project 1	http://www.rmit.edu.au/courses/039985
COSC2409	Programming Project 2	http://www.rmit.edu.au/courses/039986
COSC1284	Programming Techniques	http://www.rmit.edu.au/courses/004301
COSC1226	Real-Time Rendering & 3D Games	http://www.rmit.edu.au/courses/004227
COSC1093	Scripting Language Programming	http://www.rmit.edu.au/courses/014048

INTE1071	Secure Electronic Commerce	http://www.rmit.edu.au/courses/004178
INTE2402	Secure Programming Environment	http://www.rmit.edu.au/courses/038407
ISYS1118	Software Engineering Fundamentals	http://www.rmit.edu.au/courses/004309
COSC2299	Software Engineering Process and Tools	http://www.rmit.edu.au/courses/014049
COSC2412	Unix Essentials	http://www.rmit.edu.au/courses/039989
COSC1133	Unix Systems Administration	http://www.rmit.edu.au/courses/014050
COSC1221	User Interface Programming	http://www.rmit.edu.au/courses/004225
ISYS1126	Web Database Applications	http://www.rmit.edu.au/courses/004312
COSC2276	Web Development Technologies	http://www.rmit.edu.au/courses/035218
COSC2413	Web Programming	http://www.rmit.edu.au/courses/039990
COSC1301	Web Servers and Web Technology	http://www.rmit.edu.au/courses/004318
INTE1113	Web3D Technologies	http://www.rmit.edu.au/courses/014664
COSC2424	Windows Systems Administration	http://www.rmit.edu.au/courses/039991