TABLE OF CONTENTS

1. General Course Information

1.1 Course Details

COURSE CODE	1806ICT
COURSE TITLE	Programming Fundamentals
ACADEMIC ORGANISATION	ICT School of Information and Communication Technology
TRIMESTER	Trimester 1 2020
MODE	In Person
LEVEL	Undergraduate
LOCATION	Gold Coast, On Campus
CREDIT POINT VALUE	10

Course Description:

This is a foundational course that thoroughly covers all of the fundamental programming concepts, basic data structures, algorithmic processes, pointers and addressing, dynamic and self-referential data structures, memory management, IO, event handling, preprocessing and includes those skills and concepts that are essential to programming practice independent of underlying paradigms. The widely deployed C language is used in the context of programming embedded devices. A Co-requisite of 1010ENG Engineering Mathematics or 1017SCG Foundation Mathematics applies to students that have not obtained a grade of SA over 4 semesters in Mathematics B (Qld high school subject) in place of the first year listed elective.

Assumed Background:

Students are assumed to have obtained a grade of SA over 4 semesters in Mathematics B (Qld high school subject) or equivalent.

All students are required to have successfully completed induction for Griffith Sciences Laboratory Induction prior to commencement of practical activities. While there is a requirement for prior mathematical training, there is no requirement for prior training or experience in computer programming.

1.2 Course Introduction

Computer programming is a core skill required of computer scientists and software developers. This course introduces all of the fundamental concepts of the imperative programming paradigm: basic data structures; algorithmic processes; pointers and addressing; dynamic and self-referential data structures; memory management; input and output; event handling; and compiler preprocessing. The widely deployed C language is used as the teaching language. The C language is available to program most computer systems, including personal computers and embedded devices. The underlying concepts, skills, practices, algorithms, and the problem-solving required for programming in C are transferrable to any other programing environment. Many of the industrially popular languages have been influenced by and/or are extensions to the C language, so C is a good place to start.

Previous Student Feedback

Student feedback indicated that the the assignment should have a higher weighting. This has been done.

1.3 Course Staff

Primary Convenor **Dr Wayne Pullan**

PHONE	0755529002
EMAIL	w.pullan@griffith.edu.au
CAMPUS	Gold Coast Campus
BUILDING	Engineering (G09)
ROOM	1.66
CONSULTATION	Arrange by email.

1.4 Timetable

Timetables are available on *the Programs and Courses website*.

NB: Details contained in this section of the course profile and section 4.1 Learning Activities are to be read in conjunction with the official class timetable. The published class timetable which is the authoritative source for timetabling information for all campuses can be located by clicking on the link above.

1.5 Lecture Capture

It is standard practice at Griffith University that lectures timetabled in lecture capture-enabled venues are recorded and made available to students on the relevant course site, in accordance with the University's Lecture Capture Policy.

The lecture series delivered as part of this course will be recorded and accessible via the Learning@Griffith course site.

2. Aims, Outcomes & Graduate Attributes

2.1 Course Aims

All students enrolled in the Computer Science degree need to have a sound understanding of fundamental programming principles, and the ability to write effective and efficient computer programs. Through this course, students will be introduced to fundamental programming concepts, basic data structures, algorithmic processes, pointers and addressing, dynamic and self-referential data structures, memory management, IO, event handling, preprocessing, and building reliable programs.

2.2 Learning Outcomes

After successfully completing this course you should be able to:

1 Design, implement, test, and debug programs that use each of the following fundamental programming constructs: basic computation, device I/O, preprocessor directives, primitive variables and compound data structures, standard conditional and iterative control structures, static variables, functions, recursion, and parameter passing.

2 Design, implement, test, and debug programs that use pointers, arrays and dynamic memory management to create dynamic and self-referential data structures such as linked lists.

3 Understand the costs and benefits of dynamic and static data structure implementations, and write programs that use abstract data types.

4 Be able to develop reliable software by understanding the characteristics of reliable software and by applying a variety of strategies for testing and debugging programs.

5 Apply consistent documentation and program style standards that contribute to the readability and maintainability of software.

2.3. Graduate Attributes

For further details on the Griffith Graduate please click here

Griffith University prepares influential graduates to be:

- Knowledgeable and skilled, with critical judgement
- Effective communicators and collaborators
- Innovative, creative and entrepreneurial
- Socially responsible and engaged in their communities
- Culturally capable when working with First Australians
- Effective in culturally diverse and international environments

This table demonstrates where each of the Griffith Graduate Attributes is taught, practised and assessed in this course.

For further details on the Griffith Graduate Attributes please refer to The Griffith Graduate policy.

University wide attributes

GRADUATE ATTRIBUTE	TAUGHT	PRACTISED	ASSESSED
Knowledgeable and skilled, with critical judgement	•	•	•
Effective communicators and collaborators	•	•	•
Innovative, creative and entrepreneurial		•	•
Effective in culturally diverse and international environments		•	

Additional Course Information on Graduate Attributes

Course Program Learning Outcomes

1. An ability to apply knowledge and understanding of a widely deployed, general purpose programming language to create robust and non-trivial computer programs that solve specific problems on a variety of platforms.

Australian Computer Society (ACS) Accreditation Course Status

1. ACS Core Body of Knowledge Mappings : Bloom's Levels

Abstraction (Comprehension); Design (Application); Professional Expectations (Application); Interpersonal Communication (Application); Programming (Synthesis)

2. SFIA6 Skill :Programming / Software Development, Level 4

Designs, codes, tests, corrects, and documents moderately complex programs and program modifications from supplied specifications, using agreed standards and tools. Conducts reviews of supplied specifications, with others as appropriate.

3. ACS Complex Computing Characteristics

- *A solution requires the use of in-depth computing or domain knowledge and an analytical approach that is based on well-founded principles.*
- *Is a high-level problem possibly including many component parts or sub-problems.*
- *Identification of a requirement or the cause of a problem is ill defined or unknown.*

3. Learning Resources

3.1 Required Resources

Details of your Required Learning Resources are available from the Reading List.

3.2 Recommended Resources

Details of your Recommended Learning Resources are available from the Reading List.

3.3 University Learning Resources

The University provides many facilities and support services to assist students in their studies. Links to information about University support resources that are available to students are included below for easy reference.

Readings - New online service enabling students to access Required and Recommended Learning resources. It connects to the library catalogue to assist with quickly locating material held in Griffith libraries and enables students to manage and prioritise their readings, add personal study notes and export citations.

Learning@Griffith - there is a dedicated website for this course via the Learning@Griffith at myGriffith.

Academic Integrity Tutorial - this tutorial helps students to understand what academic integrity is and why it matters. You will be able to identify types of academic misconduct, understand what skills you will need in order to maintain academic integrity, and learn about the processes of referencing styles.

Student Support provides a range of services to support students throughout their studies including personal support such as Counselling and Health Services; Academic support; and Financial and Welfare support.

Careers and Employment Service can assist all enrolled students and recent graduates with career direction, course uncertainty, interview preparation, job search tips, LinkedIn reviews and much more. Our Unitemps Recruitment Service can assist you with finding paid casual work while you study.

Library and Learning Services: Library and Learning Services provides a wide range of quality client-focused services and programs to students, researchers and staff of the University. Library and Learning Services works in collaboration with the academic community to achieve academic and research outcomes.

Support for learning - the University provides access to common use computing facilities for educational purposes.

Code of Practice - Griffith Information Technology Resources.

3.5 Other Learning Resources & Information

Lab exercise code will be submitted electronically, and scanned with text matching software.

4. Teaching & Learning Activities

4.1 Learning Activities

Week Commencing	Activity	Learning Outcomes
24 Feb 20	**Lecture 1 (Lecture):** Introduction to Programming using C Data Types, Operators, & Expressions	1, 4, 5
2 Mar 20	**Lecture 2 (Lecture):** Flow of Control	1, 4, 5
9 Mar 20	**Lecture 3 (Lecture):** Functions	1, 4, 5
16 Mar 20	**Lecture 4 (Lecture):** Pointers	1, 2, 4, 5
23 Mar 20	**Lecture 5 (Lecture):** Arrays & Strings	1, 2, 4, 5
13 Apr 20	**Lecture 6 (Lecture):** Dynamic Memory Management	1, 2, 4, 5
27 Apr 20	**Lecture 7 (Lecture):** Bitwise Operations Input & Output	1, 4, 5
4 May 20	**Lecture 8 (Lecture):** Structures & Unions	1, 4, 5
11 May 20	**Lecture 9 (Lecture):** Data Structures	1, 2, 3, 4, 5
18 May 20	**Lecture 10 (Lecture):** Recursion	1, 4, 5
25 May 20	**Revision (Lecture):** Revision	1, 2, 3, 4, 5

4.2 Other Teaching and Learning Activities Information

- Attendance and participation at laboratory workshops is strongly recommended and expected. Students will apply concepts learnt in lectures to design and develop C programs. Lab tutors will provide guidance and help to solve the programming exercises. **Feedback is also provided in the weekly laboratory workshops**, enabling students to progress in the course.
- The lecture program will be supported by lecture notes published on the course web site. Attendance at lectures is strongly encouraged. Important course announcements may be made during the lectures and extra course content not contained in either this outline or lecture notes may also be presented.
- Students are encouraged to research and read the references and other material relevant to the course. The lecture notes are regarded as only guidelines and summaries to provide the basis for further reading. Other reference and support material will be listed on the course web-site and students are required to use their own resources to further develop their knowledge and skills.

5. Assessment Plan

5.1 Assessment Summary

Please refer to your course site in Learning@Griffith for up-to-date information regarding assessment items.

This is a summary of the assessment in the course. For detailed information on each assessment, see **5.2 Assessment Detail** below.

ASSESSMENT TASK	DUE DATE	WEIGHTING	MARKED OUT OF	LEARNING OUTCOMES	MAXIMUM EXTENSION PERIOD
Test or quiz Assessed Programming	9 Mar 20 In weeks 3, 5, 7, 9, 11 or one week later if affected by a public holiday.	20%	100 marks	1, 2, 4, 5	
Assignment - Problem Solving Assignment Project	9 Mar 20 09:00 - 15 May 20 22:00	30%	100 marks	1, 2, 4, 5	
Exam - constructed response Final Examination	Examination Period	50%	100 marks (Must achieve a min of 40 out of 100)	1, 2, 4, 5	

5.2 Assessment Detail

Title: Assessed Programming
Type: Test or quiz
Learning Outcomes Assessed: 1, 2, 4, 5
Due Date:
9 Mar 20 In weeks 3, 5, 7, 9, 11 or one week later if affected by a public holiday.
Weight: 20%
Marked out of: 100
Task Description:
The five assessed programming exercises (worth 4% each) will test student's basic problem-solving skills by requiring them to solve small problems within a given period.
Criteria & Marking:
The assessed programming tests are worth 4% each.

Unless at least 4 of these programming tests are performed, this assessment item will be considered not to have been submitted and no mark will be recorded.

Assessment feedback will be provided at the next lecture or laboratory session.

Submission: In Person at the School Department.

This assessment item:
- is a school based activity
- is an individual activity
- does not include a self assessment activity
- does not have a re-attempt provision

Title: Project
Type: Assignment - Problem Solving Assignment
Learning Outcomes Assessed: 1, 2, 4, 5
Due Date:
9 Mar 20 09:00 - 15 May 20 22:00
Weight: 30%
Marked out of: 100
Task Description:
Students are required to submit a project which will assess their ability to work on a larger software development project. The project will cover all aspects of the course.
Criteria & Marking:
30% weighting.

A detailed requirements specification will be provided on Learning@Griffith.

The assignment submission must be demonstrated and verbally explained to the tutor before any marks can be awarded for the submission.

Assessment feedback will be provided upon request two weeks after submission date.

Submission: Via the 'Assignments' tool in Learning@Griffith. Assignment will be submitted via Learning@Griffith

This assessment item:
- is a school based activity
- is an individual activity
- does not include a self assessment activity
- does not have a resubmission provision

Title: Final Examination
Type: Exam - constructed response
Learning Outcomes Assessed: 1, 2, 4, 5
Due Date:
Examination Period
Weight: 50%
Marked out of: 100
Duration: 120 minutes
Format: Online
Task Description:
The final examination will test students' knowledge and skills acquired during the course.
Criteria & Marking:
The final exam will be closed book and students will not be allowed any written material. Relevant information will be provided in the exam. Calculators will not be allowed.

Assessment feedback will be provided upon request two weeks after submission date.

This assessment item:
- is a centrally organised activity
- is an individual activity
- does not include a self assessment activity
- is a non-standard duration examination
- contains a mandatory pass component - refer to section 5.4 for further details

5.3 Late Submission

An assessment item submitted after the due date, without an approved extension, will be penalised. The standard penalty is the reduction of the mark allocated to the assessment item by 5% of the total weighted mark for the assessment item, for each working day that the item is late. A working day will be defined as Monday to Friday. Assessment items submitted more than five working days after the due date will be awarded zero marks. To understand how the mark is reduced please refer to Assessment Submission and Return Procedures

5.4 Other Assessment Information

Griffith University Disclosure Statement

The University shall provide reasonable adjustments to assessment for students with disabilities consistent with the Disabilities Standards for Education 2005, while maintaining the academic integrity of its programs. Adjustments shall be made on an individual basis. Please refer to this policy as it sets out the principles and processes that guide the University Reasonable Adjustments for Assessment - Students with Disabilities

Supplementary Assessment is available in this course in accordance with Section 8 of the University Assessment Policy. To achieve a Pass grade for the course a pass mark for the supplementary assessment item must be achieved. Students are required to submit all assessment items for this course to be eligible for a supplementary assessment.

Final Grades

A student's final grade for this course will be based on the aggregation and weighting of marks across assessment, any mandatory pass components and grade cut-offs. Grade cut-offs can vary, so you will need to wait for the official release of grades to be sure of your grade for this course.

- This course is a graded course (i.e 7, 6, 5, 4, 3, 2, 1).
- This course contains mandatory pass components.

Mandatory pass component

To be eligible to pass this course, students must:

1. achieve an overall pass mark for this course
2. achieve a minimum percentage mark of 40% [min 40 out of 100 for 'Final Examination']

6. Policies & Guidelines

This section contains the details of and links to the most relevant policies and course guidelines. For further details on University Policies please visit the Policy Library

6.1 Assessment Related Policies and Guidelines

University Policies & Guidelines

The University's assessment-related policies can be found in the Griffith Policy Library.

The Assessment policy covers topics including: assessment requirements; award of grades; supplementary assessment; special consideration; extensions and deferred assessment; conduct of students in examinations; cheating; plagiarism; notification of results; appeals against the award of grades.

Academic Integrity

Student academic misconduct encompasses all behaviour:

- involving the misrepresentation of academic achievement; or
- undermining the core values (honesty, trust, fairness, respect and responsibility) of academic integrity; or
- breaching academic integrity;

whether intentional or unintentional. Student academic misconduct includes doing as well as attempting to do any of the acts, omissions or things that constitute academic misconduct.

Student academic misconduct is defined in the Institutional Framework for Promoting Academic Integrity among Students.

Please also refer to the Student Academic Misconduct Policy.

Reasonable Adjustments for Assessment - Students with Disabilities Policy

The Reasonable Adjustments for Assessment - Students with Disabilities Policy sets out the principles and processes that guide the University in making reasonable adjustments to assessment for students with disabilities while maintaining the academic integrity of its programs.

Griffith University Disclosure Statement

The Griffith University Disclosure Statement has been developed to identify and negotiate whether necessary and reasonable accommodations and adjustments can be made, wherever possible, to enable students with disabilities and/or health conditions to undertake required learning activities. Course Convenors are encouraged to reference the Griffith University Disclosure Statement in the Learning Activities and Assessment Plan sections of their course profiles.

Assessment, how to submit an assignment and exams, viewing your grades

All you need to know about assessment, exams and grades

Text Matching Software

The University uses text matching software. Students should be aware that your Course Convenor may use software to check submitted assessment tasks. If this is the case, your Course Convenor will provide more detailed information about how the software will be used for individual assessment items.

Related links:

- Academic Integrity website
- Academic Standing, Progression and Exclusion Policy
- Assessment Policy
- Assessment Submission and Return Procedures
- End of Trimester Centrally Administered Examinations Policy and Procedures
- Governance of Assessment and Academic Achievement Standards
- Standards for First Year Assessment
- Institutional Framework for Promoting Academic Integrity among Students
- Student Academic Misconduct Policy

6.2 Other Policies and Guidelines

University Policies and Guidelines

Students are responsible for ensuring that they have read all sections of the Course Profile for the course/s in which they are enrolled in any enrolment period. The published online version of the Course Profile is the authoritative version and by the publication of the Course Profile online, the University deems the student has been notified of and read the course requirements. Variations to the Course Profile during the trimester of offer are not permitted except in exceptional circumstances and will be advised in writing to all enrolled students and via the *Learning@Griffith* website. Additional information regarding the content of this course may be published on the *Learning@Griffith* website.

Copyright matters

Copyright applies to all teaching materials and materials generated by students which substantially relate to Griffith University courses. *Students are warned against selling Griffith University teaching materials and their student notes online through commercial websites during and after their studies*. You will almost certainly be in breach of copyright law and Griffith's IT Code of Practice if you post these materials on the internet and commercial websites. Please refer to the Copyright Guide for Students for further information.

Health and Safety

Griffith University is committed to providing a safe work and study environment. However, all students, staff and visitors have an obligation to ensure the safety of themselves and those whose safety may be affected by their actions. Staff in control of learning activities will ensure as far as reasonably practical, that those activities are safe and that all safety obligations are being met. Students are required to comply with all safety instructions and are requested to report safety concerns to the University.

General health and safety information is available on the Health, Safety and Wellbeing website.

Other Key Student-Related Policies

All University policy documents are accessible to students via the Griffith Policy Library and links to key policy documents, in addition to those listed in 6.1 above, are included below for easy reference:

- Student Communications Policy
- Health and Safety Policy
- Student Administration Policy
- Student Charter
- Student Review and Appeals Policy
- Student Review and Appeals Procedures
- Student Complaints Policy

Other Course Guidelines

Students should refer to the course's Learning@Griffith site for further information about this course.

Learning Summary

Below is a table showing the relationship between the learning outcomes for this course, the learning activities used to develop each outcome and the assessment task used to assess each outcome.

Learning Outcomes

After successfully completing this course you should be able to:

1 Design, implement, test, and debug programs that use each of the following fundamental programming constructs: basic computation, device I/O, preprocessor directives, primitive variables and compound data structures, standard conditional and iterative control structures, static variables, functions, recursion, and parameter passing.
2 Design, implement, test, and debug programs that use pointers, arrays and dynamic memory management to create dynamic and self-referential data structures such as linked lists.
3 Understand the costs and benefits of dynamic and static data structure implementations, and write programs that use abstract data types.
4 Be able to develop reliable software by understanding the characteristics of reliable software and by applying a variety of strategies for testing and debugging programs.
5 Apply consistent documentation and program style standards that contribute to the readability and maintainability of software.

Assessment & Learning Activities

LEARNING ACTIVITIES	LEARNING OUTCOMES				
	1	2	3	4	5
Lecture 1 (Lecture)	●			●	●
Lecture 2 (Lecture)	●			●	●
Lecture 3 (Lecture)	●			●	●
Lecture 4 (Lecture)	●	●		●	●
Lecture 5 (Lecture)	●	●		●	●
Lecture 6 (Lecture)	●	●		●	●
Lecture 7 (Lecture)	●			●	●
Lecture 8 (Lecture)	●			●	●
Lecture 9 (Lecture)	●	●	●	●	●
Lecture 10 (Lecture)	●			●	●
Revision (Lecture)	●	●	●	●	●
ASSESSMENT TASKS					
Assessed Programming	●	●		●	●
Project	●	●		●	●
Final Examination	●	●		●	●

Graduate Attributes

For further details on the Griffith Graduate please click here

Griffith University prepares influential graduates to be:

- Knowledgeable and skilled, with critical judgement
- Effective communicators and collaborators
- Innovative, creative and entrepreneurial
- Socially responsible and engaged in their communities
- Culturally capable when working with First Australians
- Effective in culturally diverse and international environments

This table demonstrates where each of the Griffith Graduate Attributes is taught, practised and assessed in this course.

University wide attributes

GRADUATE ATTRIBUTE	TAUGHT	PRACTISED	ASSESSED
Knowledgeable and skilled, with critical judgement	•	•	•
Effective communicators and collaborators	•	•	•
Innovative, creative and entrepreneurial		•	•
Socially responsible and engaged in their communities			
Culturally capable when working with First Australians			
Effective in culturally diverse and international environments		•	