Creative Coding 1701ICT - Tri 2 2019 - Nathan Campus - In Person

1. General Course Information

1.1 Course Details

Course code	1701ICT
Course title	Creative Coding
Academic organisation	ICT School of Information and Communication Technology
Trimester	Trimester 2 2019
Mode	In Person
Level	Undergraduate
Location	Nathan, On Campus
Credit point value	10

Course Description:

Coding is an important part of today's world. In this course you will learn how to code in a creative context, utilising a practical and hands-on approach, producing generative art, data visualisations, and interactive interfaces.

Assumed Background:

This is an introductory course and requires no prior programming knowledge.

1.2 Course Introduction

In this course you will learn how to code by producing interactive art. We will cover concepts such as the drawing plane and drawing simple shapes, responding to user input, understanding physics required to produce simple simulations, applying filters to video and images in real-time, sound, and the third dimension. At the end of this course you will have a strong understanding of the fundamentals of coding as well as the ability to apply them to generative art, data visualisation, and interactive animations.

Previous Student Feedback

Previous students found the course to be an interesting, challenging, and fun introduction to coding. They enjoyed the visual and artistic freedom given to the students.

1.3 Course Staff

Primary Convenor **Dr Jolon Faichney**

Phone	(07) 555 28792
Email	j.faichney@griffith.edu.au
Campus	Gold Coast Campus
Building	Engineering (G09)
Room	2.13
Other Location	1.68

Campus Convenor
Dr Andrew Rock

Phone	(07) 373 55016
Email	a.rock@griffith.edu.au
Campus	Nathan Campus

Building	Technology (N44)

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

This course introduces practical computer programming concepts and skills through creative ideas and challenges. Students will develop programs that can generate images, animations and sound and learn how to interact with them to change their behavior. Starting with simple processes, the course will develop students' programming skills by introducing algorithmic techniques for increasingly complex visual and sonic digital projects.

2.2 Learning Outcomes

After successfully completing this course you should be able to:

1 Create interactive programs by understanding the graphics pipeline and handling user events

2 Solve programming problems that involve simple programming constructs such as variables, sequence, selection, iteration, and arrays

3 Develop software which is able to transform data over the network into visualisations and sonic experiences

4 Analyse and design a modular program using functions and objects

5 Integrate visualisations into web pages using JavaScript libraries

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	•	•	•

Additional Course Information on Graduate Attributes

Course Program Learning Outcomes

Technology building - programming, human-computer interaction, systems development, and systems acquisition

Australian Computer Society (ACS) Accreditation Course Status

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

Design (Application); Programming (Application); Human Factors (Application); System Development (Application)

2. SFIA6 Skill :Programming / Software Development, Level 3

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. 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 Services 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.

4. Teaching & Learning Activities

4.1 Learning Activities

Week Commencing	Activity	Learning Outcomes
8 Jul 18	**Course Introduction (Lecture):** Course overview and introduction to interaction and generative art, and the p5.js tool.	1
15 Jul 19	**Drawing and Basic Interaction (Lecture):** Learn about co-ordinate systems, colours, and how to draw shapes. Use global variables to provide simple interactive interfaces.	1, 2

Week Commencing	Activity	Learning Outcomes
22 Jul 19	**Animation and Interaction (Lecture):** Learn about the p5.js draw loop and variables to perform simple animations. Use if statements to respond to mouse input events.	1, 2
29 Jul 19	**A Simple Game (Lecture):** Produce a simple game using images and keyboard event handling. Display text and perform simple object collision detection. Reuse code by creating your own functions.	1, 2
5 Aug 19	**Loops and Arrays (Lecture):** Use arrays to manage multiple objects.	1, 2
19 Aug 19	**Designing Software (Lecture):** Learn techniques to solve larger programming problems.	4
2 Sep 19	**Working with Libraries (Lecture):** Use a sprite library and handle multiple scenes.	2, 4
9 Sep 19	**Visualising Data (Lecture):** Read files from over the network. Learn about file formats such as CSV and JSON. Visualise data by drawing charts.	1, 3
16 Sep 19	**Image Effects, Video, and Sound (Lecture):**	1
23 Sep 19	**3D (Lecture):** Build 3D visualisations using p5.js's 3D functionality.	1
30 Sep 19	**Beyond p5.js (Lecture):** Learn to interact and manipulate DOM elements outside the p5.js canvas.	5