Introduction to Programming – part two

Course Description

This course is aimed at novice programmers who have learnt basic programming techniques in Javascript and p5.js through Introduction to Programming – part one. The course builds on this knowledge by developing learners ability to combine short segments of code to create larger projects.

Learners will begin by developing their knowledge of object oriented programming through learning about principles such as encapsulation and abstraction. The module centres around three case study applications, a data visualiser, a drawing app, and a music visualiser. The course materials outline the structure of each and guides learners to build their own features for the apps. Finally, learners will choose one of these case studies to extend it with their own functionality.

Through this work learners will understand how to organise, plan and evaluate their code.

Course Goals and Objectives

Upon successful completion of this course, you will be able to:

1. Use a range of basic programming techniques to create complete programs
2. Adapt existing code and adapt it to customise its functionality
3. Refactor code using basic object orientation to create extendable programs
4. Effectively manage the development of a project using an iterative approach
5. Interpret API documentation to make use of use third party libraries
6. Create a set of tests and use them to evaluate your software

Textbook and Readings

There is no required textbook for this course. Listed below are some books to supplement some of the material covered in this course.

Getting Started with p5.js

Lauren McCarthy, Casey Reas, and Ben Fry

The Pragmatic Programmer – Your Journey to Mastery

David Thomas, Andrew Hunt

Coders at Work: Reflections on the Craft of Programming

Peter Seibel

Course Outline

The course consists of 10 topics, each of which spans 2 weeks.

Topic 1: Object Orientation in Practice	Learning Outcomes: ● Reinforce Constructor functions learned in the 'Introduction to programming, part 1' module. ● Understand how to split code across multiple files. ● Learn and apply Object Orientation design principles.
Topic 2: Introducing Case Study - Drawing App	Learning Outcomes: ● Practice object orientation design and technique ● Interpret a larger code base ● Use the p5.dom library to create and interact with p5.dom elements. ● Use function call backs

Topic 3: Introducing Case Study - Music Visualizer	Learning Outcomes: • Practice object orientation design and technique • Interpret a larger code base • Use the p5 sound library to play and analyse soundfiles • Understand basic properties of sound waves to visualise digital sound
Topic 4: Introducing Case Study - Data Visualisation	Learning Outcomes: • Practice object orientation design and technique • Interpret a larger code base • Understand p5's data functions and use them to manipulate and visualise data
Topic 5: Extending the Case Studies – part 1	Learning Outcomes: • Extend the functionality of an exisiting codebase • Use the random function to draw patterns • Use a fast fourier transform to create a music visualisation • Source and use exisiting data to create a visualisation

Topic 6: Extending the Case Studies – part 2	Learning Outcomes: • Extend the functionality of an exisiting codebase • Use object oriented design to create editable shapes • Use FFT portions to create a music visualisation • Create and use a new data source
Topic 7: Extending the Case Studies – part 3	Learning Outcomes: • Extend the functionality of an exisiting codebase • Create a tool using pixel data • Use beat detection to create a rhythmic music visualisation • Use object oriented techniques to create an animated data visualisation
Topic 8: Asynchronous programming	Learning Outcomes: • Understand and use asynchronous function calls to write event driven code. • Use the step debugger to identify and fix code errors
Topic 9: Testing for Stability and Performance	Learning Outcomes: • Design tests to evaluate code stability

	• Use advanced debugging techniques to identify performance bottlenecks
Topic 10: Finishing your project	Learning Outcomes: • Design and test for usability • Write a project report

Learning Activities of This Course

The course is comprised of the following elements:

- Lecture videos. In each topic the concepts you need to know will be presented through a collection of short video lectures. You may stream these videos for playback within the browser by clicking on their titles or download the videos. You may also download the slides that go along with the videos.
- Practice Quizzes. Topics include practice quizzes, intended for you to assess your understanding of the content. You will be allowed unlimited attempts at each practice quiz. There is no time limit on how long you take to complete each attempt at the quiz. These quizzes do not contribute toward your final score in the class.
- Peer Reviewed Assignments. Some topics include a peer reviewed assignment. You will be asked to submit your code for the assignment. You will then be required to review three of your peers' submissions. You can attempt these assignments multiple times. Your highest score will be used when calculating your final score in the class.
- Graded Assignments. There are two graded assignments, the first is worth 30% of the final module grade and the second 70%. There is an interim project submission in topic 6, which requires you to present your progress to date through a code submission and brief report. The final submission is in topic 10. This also requires a code submission and short written report. Both assignments will be graded by the project tutors.
- Discussion Prompt. Topics also include discussion prompts. You will see the discussion prompt alongside other items in the lesson. Each prompt provides a space for you to respond. After responding, you can see and comment on your peers' responses. All prompts and responses are also accessible from the general discussion forum and the topic discussion forum.
- Readings. Topics may include several suggested readings. They are good supplementary materials for you to further understand the course topics.

How to Pass This Course

The course has two major assessments:

- Interim progress (30%): This assignment is to be submitted in week 12 of the course. It comprises a code submission of your work to date and a small written report.
- Final project submission (70%): This assignment is to be submitted in week 20 of the course. It comprises a code submission of your final project and a written report. It will be assessed at the end the course in week 22.

This is a detailed breakdown of all of the marks.

Activity	**Required?**	**Deadline week**	**Estimated time per week**	**% of final grade**
Interim progress	Yes	12	1-2 hours	30%
Final project submission	Yes	20	2 hours	70%