UNIVERSITY OF LONDON

BSc Computer Science

Module Specification

Key Information			
Module title	Introduction to Programming II		
Level	4	Credit value	15
Member Institution	Goldsmiths	Notional study hours and duration of course	150
Module lead author/ Subject matter expert	Simon Katan		
Module co-author			

Rationale for the module

This module builds on and extends the programming skill set you developed in Introduction to Programming I. You will study a collection of more extensive, case study programs and you will work on a larger programming project from a pre-existing code base. Working on pre-existing computer programs written by others is one of the most common activities for professional programmers, and this module provides you with opportunities to gain vital experience in this area. Another common activity for professional programmers that is covered in this module is the use of third-party libraries.

Aims of the module

This module is focused on adding to the basic programming skill set you developed in Introduction to Programming I and giving you experience working with existing code and third-party libraries. By taking this module, you will learn how to customise existing code, to implement basic object orientation and to work with a range of third-party libraries.

Topics covered in this module:

The topics listed here are an approximation of what will be covered. The topics presented may be slightly revised to ensure currency and relevance. Students will be advised of any changes in advance of their study.

1. Object Orientation in Practice
2. Introducing Case Study - Drawing App
3. Introducing Case Study - Music Visualizer
4. Introducing Case Study - Data Visualisation
5. Extending the Case Studies – part 1
6. Extending the Case Studies – part 2
7. Extending the Case Studies – part 3
8. Asynchronous programming
9. Testing for Stability and Performance
10. Finishing your project

Learning outcomes for the module

Students who successfully complete this module 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

Assessment strategy, assessment methods

Summative and Formative Assessments
The module will contain a range of summative and formative assessments. Summative assessments are assessments which contribute directly towards your final grade. Formative assessments do not count directly towards your final grade. Instead, they provide you with opportunities for low stakes practice, and will often provide some sort of feedback about your progress. For example, a practice quiz might provide you with feedback about why a particular answer was wrong.

This module will primarily be assessed through a substantial programming project. It will be submitted in two parts: a work in progress submission mid-session and a final submission at the end of the session. The project will be supplemented with continual assessment consisting of regular quizzes and small programming assignments.

Assessment Activities
The table below lists the assessment activity types you might encounter taking the module. It also states if that type of assessment can be automatically graded. For example, multiple choice quizzes can be automatically graded, and so can some programming assignments. It also states if that type of assessment will be found in the summative courseworks. More details about the summative assessments are provided below.

Assessment activity type	Can it be automatically graded with feedback in some cases?	CW1	CW2
Quiz	X	X	X
Writing task		X	X
Programming task	X	X	X
Peer review task		X	X

Pass Mark

In order to pass this module, you must achieve at least 35% in each element of summative assessment and an overall weighted average of 40%, subject to the application of rules for compensation. Please refer to the programme regulations for more information.

Summative Assessment Elements

This module will primarily be assessed through a substantial programming project.

Summative Assessment Component	Components	Percentage of final credit	Deadline
Coursework 1	Work in progress submission	30%	Mid session
Coursework 2	Final Project Report	70%	End of session

The assessment for the project comprises two coursework components. The first is designed to take 15 hours to complete and is a work in progress report on the programming project. The second coursework is designed to take 35 hours to complete and is the final report for the programming project, including source code.

Learning resources

The module will draw on a number of different, largely web-based, public resources as well as the resources produced as bespoke material for this module.

The programming language will be Javascript, with the p5js library used for graphical and interactive programming functionality. The main external resource will be the set of online tutorials available from:

https://p5js.org/learn/

The textbooks will be:

Zakas, N., 2014. *The Principles of Object-Oriented JavaScript*. 1st ed. CA: No Starch Press.

Crawford, C., 2002. *Art of Interactive Design: A Euphonious and Illuminating Guide to Building Successful Software*. 1st ed. CA: No Starch Press.