



Module Specification

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

Rationale for the module
<p>This module provides you with the foundational skill set required to write computer programs. Writing computer programs is a crucial activity when studying computer science, and you will be expected to be able to read and write program code at many points during your studies.</p>

Aims of the module
<p>This module is focused on basic programming techniques. By taking this module, you will learn how to use the basic elements of computer programming such as variables, conditionals, functions and loops. You will also learn how to create interactive, graphical computer programs. You will also be introduced to basic object-oriented programming techniques.</p>

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. Your development environment
2. Drawing in 2D
3. Variables, Objects and Interaction
4. Conditional and variable types
5. For loops and arrays
6. Advanced arrays and iteration
7. User defined functions and scope
8. Data structures and searches
9. Extending objects
10. Constructor functions

Approximately 10-12 hours of study will be required per topic. The remaining study time is intended for coursework.

Learning outcomes for the module

Students who successfully complete this module will be able to:

1. Create and run interactive, graphical computer programs
2. Use variables and conditionals to store and act upon changing program state
3. Initialise and access arrays of variables, and use loops to work with arrays and iteration
4. Write custom functions and event handling code
5. Use encapsulation and constructors to define basic objects
6. Debug pre-existing program code

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.

The module will include continuous assessments, primarily comprising regular summative quizzes and small programming exercises.

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 (CW1, CW2). 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

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%. This module cannot be compensated. Please refer to the programme regulations for more information.

Summative Assessment Elements

This is a module that is best assessed largely through continuous assessment by way of exercises worked on throughout the session.

Summative Assessment Component	Components	Percentage of final credit	Deadline
Coursework 1	Four programming exercise submissions	50%	Mid session
Coursework 2	Four programming exercise submissions	50%	End of session

Each of the two courseworks will take up to 25 hours of study time to complete and comprise a variety of practical exercises and quizzes.

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/>

There is no required textbook for this module.