

Introduction to programming 1

Course Description

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

Learners will be provided with the foundational skill set required to write computer programs. Writing computer programs is a crucial activity when studying computer science, and learners will be expected to be able to read and write program code at many points during their studies.

Course Goals and Objectives

Upon successful completion of this course, a learner 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.

Textbook and Readings

There is no required textbook for this module. The module draws on number of different, largely web-based, public resources as well as the resources produced as bespoke material for this module. The programming language is Javascript, with the p5js library used for graphical and interactive programming functionality. The main external resource is the set of online tutorials available from: <https://p5js.org/learn/>.

Specific readings for each topic are listed with direct links to free online recourses that provide additional material on the topics of this course.

Course Outline

The course consists of ten topics divided in 20 weeks that focus on key concepts.

Topic 1. Your development environment	<p>Key concepts: the toolkit we will use to develop programs; drawing using simple shapes and 2D coordinates; and the process and experience of writing code.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none">• Set up the programming tool kit and use it to edit and run programs.• Use built-in drawing functions and 2D coordinates to draw simple shapes on the screen.• Describe the process of writing code.
Topic 2. Drawing in 2D	<p>Key concepts: set of 2D drawing functions in p5.js; colour representation and styles; and debugging with the console.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none">• Use the drawing functions to draw lines, ellipses and rectangles.• Use the fill, noFill, background, stroke and noStroke functions to apply styles and colours to shapes.• Use the browser console to identify and fix basic syntax errors.

<p>Topic 3. Variables, objects and interaction</p>	<p>Key concepts: variables and data; simple interaction; and code organisation.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Use variables to store program state and use operators to change variables. • Use built-in variables and events to create simple interactivity. • Use appropriately named variables and objects to organise code.
<p>Topic 4. Conditionals and variable types</p>	<p>Key concepts: conditional logic and conditionals with types. You will also learn some new debugging techniques.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Use if else blocks to control the flow of a program based on a test condition. • Define test conditions involving variables and logical operators. • Use isolation and printing to identify and repair buggy code. • Design test conditions which are appropriate for different types of variable.
<p>Topic 5. For loops and arrays</p>	<p>Key concepts: the loops and using arrays to store collections of variables.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Use arrays to store multiple values. • Use a for loop to run a block of code multiple times. • Use the index within a for loop to produce graphical patterns. • Access items from an array.

	<ul style="list-style-type: none"> • Control your workflow for optimal coding.
Topic 6. Advanced arrays and iteration	<p>Key concepts: using arrays and for loops together and using nested for loops to draw grids to the canvas.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Use for loops to iterate over an array storing information about scenery. • Use string processing functions to analyse the content of strings. • Explain why coding is an iterative process involving gradual improvement of the product.
Topic 7. User defined functions and scope	<p>Key concepts: writing own functions and understand about their place in Javascript and variable scope.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Use functions with parameters and return types to create program modules with well defined purpose. • Use the concept of scope to explain how different parts of a program see different variables. • Explain why modular programming leads to better organised and easier to read code.
Topic 8. Data structures and searches	<p>Key concepts: 2D arrays that store other arrays. You will learn several coding techniques on how to find specific values in an array.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Use nested loops to iterate through 2D arrays of numbers and generate images

	<p>from them.</p> <ul style="list-style-type: none"> • Use array push and splice functions to add to and combine arrays. • Use arrays of different types of objects to generate graphics for a game level.
Topic 9. Extending objects	<p>Key concepts: adding functionality to the properties and using the P5 vector object.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Differentiate between functions, variables, methods and properties. • Implement objects with methods and properties and use them to model specified problems. • Model enemies and platforms in a video game using objects and arrays.
Topic 10. Constructor functions	<p>Key concepts: learning about the basics of object orientation and how to write applications that create their own objects using constructor functions and the factory pattern.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Define constructor functions that can be used to create and configure new objects. • Differentiate between object literals and constructor functions and select which is the best to use in a given scenario. • Use the splice function to remove objects from an array, being aware of the problems this causes with iteration.

Learning Activities of This Course

The course is comprised of the following elements:

- **Sleuth crime adventure.** This activity will teach you the fundamentals of programming. In the adventure, you are a fledgling private investigator working for Sleuth and Co detective agency. Your job is to use your programming skills to solve code crimes which are arranged into cases. During your adventure the chief will keep you on track. This assignment covers the Rookie stage of Sleuth where you can attempt to solve 9 cases. After the deadline you will go Pro and there will be another 7 cases for you to attempt. Your final grade is calculated as an average of your Rookie and Pro scores.
- **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.** Each topic will include one practice quizzes, intended for you to assess your understanding of the topics. You will be allowed unlimited attempts at each practice quiz. Each attempt may present a different selection of questions to you. 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.** There will be one peer reviewed assignment for every part of the game project. You will read a short prompt, then answer several questions about the information provided. You will then be required to review three of your peers' submissions. Peer reviewed assignments will not contribute to your final grade, but will be valuable practice for your game project.
- **Discussion Prompt.** Each topic may include one discussion prompt. 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 module discussion forum.
- **Readings.** Each topic 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 each worth 50% of your grade:

- **Coursework 1:** this consists of two activities that you do on the Coursera platform and which will be assessed half way through course (week 10 and week 12)
- **Coursework 2:** this also consists of two activities that you do on the Coursera platform and which will be assessed in the end the course in week 22.

There are also several activities that are graded but have 0 weight. That means that they will not count towards your final grade, but they are a key part of your learning and you need to do them.

This is a detailed breakdown of all of the marks.

Activity	Required?	Deadline week	Estimated time per week	% of final grade
Sleuth Rookie stage	Yes	1-10	1-2 hours	25%
The game project, parts 1-4	Yes	12	2 hour	25%
Sleuth Pro stage	Yes	22	1-2 hours	25%
The game project, parts 5-8	Yes	22	2 hours	25%