# 

# KS4 – Object-oriented programming

## Unit introduction

Although it is not on the specifications currently, the topic of object-oriented programming is a useful one to cover with your students will give students an insight into a different programming paradigm. This unit aims to take them from a rote understanding of programming concepts to a more nuanced understanding of the ways they can choose to write programs. Using a combination of relatable examples (animals, vehicles and books) and practical activities the students will gain an understanding of the concepts of objects and classes.

This unit is an ideal choice for a class you want to prepare for A-level computing or want to stretch beyond the standard programming practise. That being said, the activities can be used with any class as they have been scaffolded accordingly. Ideally, learners will have completed the KS4 programming unit before attempting this unit. The learners should be comfortable with the basic programming constructs like if statements, variables and loops.

## Overview of lessons

|  |  |  |
| --- | --- | --- |
| **Lesson** | **Brief overview** | **Learning objectives** |
| Programming paradigms | In this first lesson students will be reminded of the importance of programming conventions and their benefits to both readers and writers of programs. The activities will encourage students to think about the way they write programs. Students will explore the idea of programming paradigms, looking to procedural programming as an example of one they already use. In the last activity you will introduce the topic of this unit, object-oriented programming | * Describe the role of conventions in programming * Recall that there are different paradigms for programming * Define object oriented programming * Identify a class and object as a part of a program |
| Classes and Objects | During this lesson, students will be introduced to the concepts of classes and objects through a classification exercise to determine the properties shared by all animals. As the lesson progresses, Students will learn and apply key terminology, and have the opportunity to create an object class and instantiate that object. | * Describe the relationship between a class and an object * Model a real world problem using object oriented programming conventions. * Define attributes and methods as a part of a class * Use a constructor to create objects * Use a method and access an attribute on an object |
| Creating a class | In this lesson your students will be independently applying the learning they have been doing in the first two lessons of the unit. You will use the Pet class created in lesson two as a worked example for the students to apply in creating a new class. The students will be stepping into the role of a video game programmer, on a fictional game called *Monster Quest* they will be making a class to represent the monsters in the game. | * Create a class * Define the use of a self parameter in object-oriented Python * Create a method on a class * Access and modify attributes using getters and setters |
| Inheritance | In this lesson your students will discover and define the object-oriented principle of inheritance. They will explore this concept first by relating it back to the Pet example used in the previous lesson, as well as defining key terms such as subclass and superclass. The practical part of this lesson will involve making two new subclasses for the Monster class they created last lesson. One of which you will model and the second they will do independently. | * Define the principle of inheritance * Define the terms superclass and subclass * Select appropriate uses of inheritance * Create a subclass in a program |
| Assessment | In this final lesson of the object-oriented programming unit, students will undertake an assessment of their knowledge of the concepts and key vocabulary covered in the unit. The focus of this assessment will be a library tracker application made using OOP. Students will first explore the program and make notes on the classes used, before editing and adding to the program themselves. | * Explore a program written using OOP * Explain the key concepts of OOP |

## Progression

This unit progresses learners’ knowledge and understanding of object-oriented programming.

## Curriculum links

[**National curriculum links**](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study)

* Develop their capability, creativity, and knowledge in computer science, digital media, and information technology
* Develop and apply their analytic, problem-solving, design, and computational thinking skills

## Assessment

### Summative assessment

* Assessment - Object-oriented programming - KS4

## Subject knowledge

This unit focuses on the components that make up a computer system and their roles in computation. It also teaches learners about the generations of programming languages and how software and hardware work together in computation.

Enhance your subject knowledge to teach this unit through the following training opportunities:

### Online training courses

* [Object-oriented Programming in Python: Create Your Own Adventure Game](https://teachcomputing.org/courses/CO210/object-oriented-programming-in-python-create-your-own-adventure-game)
* [Programming 101: An Introduction to Python for Educators](https://teachcomputing.org/courses/CO207/programming-101-an-introduction-to-python-for-educators)
* [Programming 102: Think like a Computer Scientist](https://teachcomputing.org/courses/CO208/programming-102-think-like-a-computer-scientist)
* [Programming 103: Saving and Structuring Data](https://teachcomputing.org/courses/CO219/programming-103-saving-and-structuring-data)

### Face-to-face courses

* [Introduction to algorithms, programming and data in GCSE computer science](https://teachcomputing.org/courses/CP428/introduction-to-algorithms-programming-and-data-in-gcse-computer-science-remote)
* [GCSE Computer Science - developing outstanding teaching - face to face](https://teachcomputing.org/courses/CP205/gcse-computer-science-developing-outstanding-teaching-face-to-face)

Resources are updated regularly — the latest version is available at: [ncce.io/tcc](http://ncce.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](http://ncce.io/ogl).