

# Cambridge TECHNICALS LEVEL 3

# IT

Cambridge  
TECHNICALS  
2016

## MAPPING GUIDE

### Unit 15 Games Design and Prototyping

Version 1

# INTRODUCTION

Prodigy are delighted to work with OCR, a progressive Awarding Organisation, who share the ambition of providing high-quality qualifications, learning solutions that are industry-led and reliable and valid assessment. The Cambridge Technicals in IT qualifications provide 'future-ready' skills for a learner to further their ambitions, whether that is in terms of further academic study, enter an apprenticeship or as a springboard to gaining employment.

Prodigy Learning (Prodigy) is an award-winning EdTech business providing digital skills certifications and learning solutions for a range of technologies including Adobe, Autodesk and Microsoft. Established in 2000, Prodigy now have offices in Dublin, London and Sydney. Having worked closely with Microsoft since 2000, Prodigy is a Microsoft Authorised Education Gold Partner and a MS Global Training Partner supporting academic institutions utilise Microsoft Imagine Academy, Microsoft certifications and other Microsoft Education solutions.

Historically, the UK has thrived on a rich research and technology base and has been at the forefront of global technology innovation. Enthusiasing young learners about following exciting careers in science, technology, engineering and mathematics (STEM) subjects is fundamental to maintaining this success. However, currently the UK has a widely acknowledged skills gap in the pipeline of talent studying computing-related disciplines. Therefore, providing high quality, engaging and relevant qualifications that equip learners with current technical knowledge and skills is essential to encourage more young people into the computing discipline, and moreover to ensure they progress to jobs in the sector.

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

## 2. Design Algorithms

	2.1.1 Differentiate problems as easy or hard for computers to solve;	2.1.2 apply the concept of iteration;	2.1.3 create simple algorithms	2.2.1 Analyse a game and create a sequence of instructions for playing it;	2.2.2 identify an event;	2.2.3 create the code for an event in block-based editors;	2.2.4 explain the "on every frame" code and event handlers
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	2.1.1 Differentiate problems as easy or hard for computers to solve;	2.1.2 apply the concept of iteration;	2.1.3 create simple algorithms	2.2.1 Analyse a game and create a sequence of instructions for playing it;	2.2.2 identify an event;	2.2.3 create the code for an event in block-based editors;	2.2.4 explain the "on every frame" code and event handlers
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	2.3.1 Describe animation that uses a series of individual frames;	2.3.2 resolve errors in algorithms;	2.3.3 create algorithms that can be translated into pseudocode or block-based code;	2.3.4 use code to command items on the screen or device	2.4.1 Define loops;	2.4.2 identify the control variable;	2.4.3 predict the output of loop, random number, and control variable constructs;	2.4.4 identify conditional statements;	2.4.5 choose the appropriate Boolean logic for specific results
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/ conditional)	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/ procedures)	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	2.3.1 Describe animation that uses a series of individual frames;	2.3.2 resolve errors in algorithms;	2.3.3 create algorithms that can be translated into pseudocode or block-based code;	2.3.4 use code to command items on the screen or device	2.4.1 Define loops;	2.4.2 identify the control variable;	2.4.3 predict the output of loop, random number, and control variable constructs;	2.4.4 identify conditional statements;	2.4.5 choose the appropriate Boolean logic for specific results
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

## 3. Work with Data Representation

	3.1.1 Create code to add and position objects, such as sprites on a screen or device;	3.1.2 identify data examples as text, sound, pictures, or numbers;	3.1.3 change the parameters of "set frame grid" to work with different sprite sheets; explain the role of cloud variables;	3.1.4 explain the impact of variable scope, including cloud variables, global variables, and local or temporary variables;	3.1.5 design, create, and populate a table or two-dimensional array;	3.1.6 describe multiple uses for data.
1.1.7 Game design principles - sprites	X					
1.2.1 Game prototype - types of prototyping e.g. visual/representational (non-working) and proof of concept (working)	X		X			
2.1.1 Game elements - navigation	X	X	X	X	X	X
2.1.2 Game elements - scoring	X	X	X	X	X	X
2.1.3 Game elements - movement	X	X	X	X	X	X
2.1.4 Game elements - interaction/controls	X	X	X	X	X	X
2.1.5 Game elements - conveying information	X	X	X	X	X	X
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	3.1.1 Create code to add and position objects, such as sprites on a screen or device;	3.1.2 identify data examples as text, sound, pictures, or numbers;	3.1.3 change the parameters of "set frame grid" to work with different sprite sheets; explain the role of cloud variables;	3.1.4 explain the impact of variable scope, including cloud variables, global variables, and local or temporary variables;	3.1.5 design, create, and populate a table or two-dimensional array;	3.1.6 describe multiple uses for data.
3.2.4 Programming techniques - outputs	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines (e.g. functions/procedures)	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X



# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	3.2.1 Declare and use variables in a program;	3.2.2 use input and variables to calculate new information;	3.2.3 describe arrays, lists, and collections;	3.2.4 explain the differences between variables and arrays	3.3.1 Name your tenant;	3.3.2 set up your first administrator;	3.3.3 determine tenant location.
1.2.1 Game prototype - types of prototyping e.g. visual/ representational (non-working) and proof of concept (working)					X	X	X
2.1.1 Game elements - navigation	X	X	X	X	X	X	X
2.1.2 Game elements - scoring	X	X	X	X	X	X	X
2.1.3 Game elements - movement	X	X	X	X	X	X	X
2.1.4 Game elements - interaction/ controls	X	X	X	X	X	X	X
2.1.5 Game elements - conveying information	X	X	X	X	X	X	X
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	3.2.1 Declare and use variables in a program;	3.2.2 use input and variables to calculate new information;	3.2.3 describe arrays, lists, and collections;	3.2.4 explain the differences between variables and arrays	3.3.1 Name your tenant;	3.3.2 set up your first administrator;	3.3.3 determine tenant location.
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

## 4. Solve Computational Problems

	4.1.1 Use algorithms and Boolean logic;	4.1.2 use games and apps to simulate practical tasks such as converting currencies.	4.2.1 Create programs and apps that mimic random occurrences;	4.2.2 create programs and apps that demonstrate fundamentals of physics such as gravity, acceleration, and bounce.
1.2.1 Game prototype - types of prototyping e.g. visual/representational (non-working) and proof of concept (working)	X	X	X	X
2.1.1 Game elements - navigation	X	X	X	X
2.1.2 Game elements - scoring	X	X	X	X
2.1.3 Game elements - movement	X	X	X	X
2.1.4 Game elements - interaction/controls	X	X	X	X
2.1.5 Game elements - conveying information	X	X	X	X
2.1.9 Game elements - problems solving	X	X	X	X
3.2.1 Programming techniques - variable constants	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	4.1.1 Use algorithms and Boolean logic;	4.1.2 use games and apps to simulate practical tasks such as converting currencies.	4.2.1 Create programs and apps that mimic random occurrences;	4.2.2 create programs and apps that demonstrate fundamentals of physics such as gravity, acceleration, and bounce.
3.2.5 Programming techniques - sequence	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

## 5. Code Programs in Block-Based Programming

	5.1.1 Identify the basic coding elements of the programming environment;	5.1.2 create code for conditional statements;	5.1.3 create conditional statements using and, or, and not;	5.1.4 create loops; use counting variables	5.2.1 Code by using ready-made functions related to objects including the game board, wall, obstacle, sprite, string, teXtbox, collection, and turtle.	5.3.1 Plot points on a gaming coordinate system;	5.3.2 analyse a game to discover how random numbers are used;	5.3.3 use the random range function to generate random numbers;	5.3.4 evaluate random numbers by using conditionals
1.2.1 Game prototype - types of prototyping e.g. visual/representational (non-working) and proof of concept (working)	X	X	X	X	X	X	X	X	X
2.1.1 Game elements - navigation	X	X	X	X	X	X	X	X	X
2.1.2 Game elements - scoring	X	X	X	X	X	X	X	X	X
2.1.3 Game elements - movement	X	X	X	X	X	X	X	X	X
2.1.4 Game elements - interaction/controls	X	X	X	X	X	X	X	X	X
2.1.5 Game elements - conveying information	X	X	X	X	X	X	X	X	X
2.1.9 Game elements - problems solving	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	5.1.1 Identify the basic coding elements of the programming environment;	5.1.2 create code for conditional statements;	5.1.3 create conditional statements using and, or, and not;	5.1.4 create loops; use counting variables	5.2.1 Code by using ready-made functions related to objects including the game board, wall, obstacle, sprite, string, teXtbox, collection, and turtle.	5.3.1 Plot points on a gaming coordinate system;	5.3.2 analyse a game to discover how random numbers are used;	5.3.3 use the random range function to generate random numbers;	5.3.4 evaluate random numbers by using conditionals
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG BLOCK-B 98-380

	5.1.1 Identify the basic coding elements of the programming environment;	5.1.2 create code for conditional statements;	5.1.3 create conditional statements using and, or, and not;	5.1.4 create loops; use counting variables	5.2.1 Code by using ready-made functions related to objects including the game board, wall, obstacle, sprite, string, teXtbox, collection, and turtle.	5.3.1 Plot points on a gaming coordinate system;	5.3.2 analyse a game to discover how random numbers are used;	5.3.3 use the random range function to generate random numbers;	5.3.4 evaluate random numbers by using conditionals
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X

## 7. Examine the Software Development Process

	7.1.1 Analyse problems in relation to your audience and identify which apps or games can be part of the solution and how they can be used;	7.1.2 describe user-experience principles;	7.1.3 gather user input;	7.1.4 use code or text to create instructions for using a program.
1.2.1 Game prototype - types of prototyping e.g. visual/representational (non-working) and proof of concept (working)	X	X	X	X



## 1. Data Types and Operators

	1. Data Types and Operators	1.1 Evaluate the data types of variables, inputs, and outputs  *Data types include str, int, long, float, and Boolean	1.2.1 Type casting;	1.2.2 constructing sequence types using multiple values;	1.2.3 indexing and slicing operations	1.3.1 Assignment;	1.3.2 Comparison;	1.3.3 Logical;	1.3.4 Arithmetic;	1.3.5 Identity (is);	1.3.6 Containment (in)
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	1. Data Types and Operators	1.1 Evaluate the data types of variables, inputs, and outputs  *Data types include str, int, long, float, and Boolean	1.2.1 Type casting;	1.2.2 constructing sequence types using multiple values;	1.2.3 indexing and slicing operations	1.3.1 Assignment;	1.3.2 Comparison;	1.3.3 Logical;	1.3.4 Arithmetic;	1.3.5 Identity (is);	1.3.6 Containment (in)
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	1.4.1 Assignment;	1.4.2 Comparison;	1.4.3 Logical;	1.4.4 Arithmetic;	1.4.5 Identity (is);	1.4.6 Containment (in)
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/ conditional)	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/ procedures)	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	1.4.1 Assignment;	1.4.2 Comparison;	1.4.3 Logical;	1.4.4 Arithmetic;	1.4.5 Identity (is);	1.4.6 Containment (in)
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

## 2. Control flow (Decisions and Loops)

	2.1.1 if;	2.1.2 elif;	2.1.3 else;	2.1.4 nested and compound conditionals	2.2.1 while;	2.2.2 for;	2.2.3 break;	2.2.4 continue;	2.2.5 nested loops and loops that include compound conditionals
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	2.1.1 if;	2.1.2 elif;	2.1.3 else;	2.1.4 nested and compound conditionals	2.2.1 while;	2.2.2 for;	2.2.3 break;	2.2.4 continue;	2.2.5 nested loops and loops that include compound conditionals
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

## 3. Input and Output

	3.1.1 open;	3.1.2 close;	3.1.3 read;	3.1.4 write;	3.1.5 append;	3.1.6 check existence;	3.1.7 delete;	3.1.8 with statement	3.2.1 Read input from console;	3.2.2 print formatted text;	3.2.3 use of command line arguments
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	3.1.1 open;	3.1.2 close;	3.1.3 read;	3.1.4 write;	3.1.5 append;	3.1.6 check existence;	3.1.7 delete;	3.1.8 with statement	3.2.1 Read input from console;	3.2.2 print formatted text;	3.2.3 use of command line arguments
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X	X	X



# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

## 4. Structure and Functions

	4.1.1 Use of indentation and white space;	4.1.2 comments and documentation strings;	4.1.3 pydoc	4.2.1 Call signatures;	4.2.2 default values;	4.2.3 return;	4.2.4 def;	4.2.5 pass
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	4.1.1 Use of indentation and white space;	4.1.2 comments and documentation strings;	4.1.3 pydoc	4.2.1 Call signatures;	4.2.2 default values;	4.2.3 return;	4.2.4 def;	4.2.5 pass
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X

## 5. Troubleshooting and Error Handling

	5.1.1 Syntax errors;	5.1.2 logic errors;	5.1.3 runtime errors	5.2.1 Try;	5.2.2 except;	5.2.3 else;	5.2.4 finally;	5.2.5 raise
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	5.1.1 Syntax errors;	5.1.2 logic errors;	5.1.3 runtime errors	5.2.1 Try;	5.2.2 except;	5.2.3 else;	5.2.4 finally;	5.2.5 raise
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X

## 6. Modules and Tools

	6.1.1 math;	6.1.2 datetime;	6.1.3 io;	6.1.4 sys;	6.1.5 os;	6.1.6 os.path;	6.1.7 random	6.2.1 math;	6.2.2 datetime;	6.2.3 random
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA INTRODUCTION TO PROG PYTHON 98-381

	6.1.1 math;	6.1.2 datetime;	6.1.3 io;	6.1.4 sys;	6.1.5 os;	6.1.6 os.path;	6.1.7 random	6.2.1 math;	6.2.2 datetime;	6.2.3 random
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

## 1. JavaScript Fundamentals (operators, variables)

	1.1.1 Assignment;	1.1.2 increment;	1.1.3 decrement;	1.1.4 addition;	1.1.5 subtraction;	1.1.6 division;	1.1.7 multiplication;	1.1.8 modulus;	1.1.9 compound assignment operators
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/ conditional)	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/ procedures)	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	1.1.1 Assignment;	1.1.2 increment;	1.1.3 decrement;	1.1.4 addition;	1.1.5 subtraction;	1.1.6 division;	1.1.7 multiplication;	1.1.8 modulus;	1.1.9 compound assignment operators
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X



# MAPPED TO MTA JAVA SCRIPT 98-382

	1.2.1 Comments;	1.2.2 indentations;	1.2.3 naming conventions;	1.2.4 no script;	1.2.5 constants;	1.2.6 reserved keywords;	1.2.7 debugger keyword;	1.2.8 setting breakpoints;	1.2.9 console. log
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/ conditional)	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/ procedures)	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	1.2.1 Comments;	1.2.2 indentations;	1.2.3 naming conventions;	1.2.4 no script;	1.2.5 constants;	1.2.6 reserved keywords;	1.2.7 debugger keyword;	1.2.8 setting breakpoints;	1.2.9 console. log
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	1.3 Implement inline and external scripts. When to use, how to use, and what happens when both are used	1.4.1 try;	1.4.2 catch;	1.4.3 finally	1.5.1 Manage state;	1.5.2 display dialogs;	1.5.3 determine screen size
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	1.3 Implement inline and external scripts. When to use, how to use, and what happens when both are used	1.4.1 try;	1.4.2 catch;	1.4.3 finally	1.5.1 Manage state;	1.5.2 display dialogs;	1.5.3 determine screen size
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

## 2. Variables, Data Types, and Functions

	2.1.1 Number;	2.1.2 Boolean;	2.1.3 String; Null;	2.1.4 Undefined;	2.1.5 Ntypeof operator;	2.1.6 type checking functions;	2.1.7 use strict;	2.1.8 converting between data types;	2.1.9 formatting numbers;	2.1.10 string operations;	2.1.11 single quote vs double quote (nesting);	2.1.12 initialization
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/ conditional)	X	X	X	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/ procedures)	X	X	X	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	2.1.1 Number;	2.1.2 Boolean;	2.1.3 String; Null;	2.1.4 Undefined;	2.1.5 Ntypeof operator;	2.1.6 type checking functions;	2.1.7 use strict;	2.1.8 converting between data types;	2.1.9 formatting numbers;	2.1.10 string operations;	2.1.11 single quote vs double quote (nesting);	2.1.12 initialization
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	2.2.1 iteration;	2.2.3 initialization;	2.2.4 array definition;	2.2.5 sorting and searching;	2.2.6 push and pop;	2.2.7 shift/unshift;	2.2.8 length;	2.2.9 accessing an element;	2.2.10 understanding multi-dimensional arrays	2.3.1 Properties;	2.3.2 methods;	2.3.3 instantiation;	2.3.4 Date object;	2.3.5 time;	2.3.6 retrieving date parts;	2.3.7 localization (MM/DD vs DD/MM);	2.3.8 adding and subtracting dates
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	2.2.1 iteration;	2.2.3 initialization;	2.2.4 array definition;	2.2.5 sorting and searching;	2.2.6 push and pop;	2.2.7 shift/unshift;	2.2.8 length;	2.2.9 accessing an element;	2.2.10 understanding multi-dimensional arrays	2.3.1 Properties;	2.3.2 methods;	2.3.3 instantiation;	2.3.4 Date object;	2.3.5 time;	2.3.6 retrieving date parts;	2.3.7 localization (MM/DD vs DD/MM);	2.3.8 adding and subtracting dates
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X



# MAPPED TO MTA JAVA SCRIPT 98-382

	2.2.1 iteration;	2.2.3 initialization;	2.2.4 array definition;	2.2.5 sorting and searching;	2.2.6 push and pop;	2.2.7 shift/unshift;	2.2.8 length;	2.2.9 accessing an element;	2.2.10 understanding multi-dimensional arrays	2.3.1 Properties;	2.3.2 methods;	2.3.3 instantiation;	2.3.4 Date object;	2.3.5 time;	2.3.6 retrieving date parts;	2.3.7 localization (MM/DD vs DD/MM);	2.3.8 adding and subtracting dates
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	2.4.1 Random;	2.4.2 round; abs; floor; ceiling; min/ max; pow; sqrt	2.4.3 abs;	2.4.4 floor;	2.4.5 ceiling;	2.4.6 min/ max;	2.4.7 pow;	2.4.8 sqrt	2.5.1 Reusable code;	2.5.2 local vs global scope, redefining variables, passing parameters, value vs. reference, return values
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/ conditional)	X	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/ procedures)	X	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	2.4.1 Random;	2.4.2 round; abs; floor; ceiling; min/ max; pow; sqrt	2.4.3 abs;	2.4.4 floor;	2.4.5 ceiling;	2.4.6 min/ max;	2.4.7 pow;	2.4.8 sqrt	2.5.1 Reusable code;	2.5.2 local vs global scope, redefining variables, passing parameters, value vs. reference, return values
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X	X
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

## 3. Implement and analyse decisions and loops

	3.1 Evaluate expressions that use logical and comparison operators	3.2.1 if;	3.2.2 else if;	3.2.3 switch;	3.2.4 nested if	3.3.1 for;	3.3.2 while;	3.3.3 do;	3.3.4 break;	3.3.5 continue
3.2.1 Programming techniques - variable constants	X	X	X	X	X	X	X	X	X	X
3.2.2 Programming techniques - strings	X	X	X	X	X	X	X	X	X	X
3.2.3 Programming techniques - inputs	X	X	X	X	X	X	X	X	X	X
3.2.4 Programming techniques - outputs	X	X	X	X	X	X	X	X	X	X
3.2.5 Programming techniques - sequence	X	X	X	X	X	X	X	X	X	X
3.2.6 Programming techniques - selection	X	X	X	X	X	X	X	X	X	X
3.2.7 Programming techniques - iteration (e.g. counting/conditional)	X	X	X	X	X	X	X	X	X	X
3.2.8 Programming techniques - sub routines e.g. functions/procedures)	X	X	X	X	X	X	X	X	X	X
3.2.9 Programming techniques - conditions	X	X	X	X	X	X	X	X	X	X
3.2.10 Programming techniques - counting	X	X	X	X	X	X	X	X	X	X

# MAPPED TO MTA JAVA SCRIPT 98-382

	3.1 Evaluate expressions that use logical and comparison operators	3.2.1 if;	3.2.2 else if;	3.2.3 switch;	3.2.4 nested if	3.3.1 for;	3.3.2 while;	3.3.3 do;	3.3.4 break;	3.3.5 continue
3.2.11 Programming techniques - totalling	X	X	X	X	X	X	X	X	X	X
3.2.12 Programming techniques - data structures	X	X	X	X	X	X	X	X	X	X
3.2.13 Programming techniques - file handling (e.g. arrays analysis)	X	X	X	X	X	X	X	X	X	X
3.2.14 Programming techniques - maintain code	X	X	X	X	X	X	X	X	X	X
3.2.15 Programming techniques - libraries	X	X	X	X	X	X	X	X	X	X



We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

Whether you already offer OCR qualifications, are new to OCR, or are considering switching from your current provider/awarding organisation, you can request more information by completing the Expression of Interest form which can be found here:

[www.ocr.org.uk/expression-of-interest](http://www.ocr.org.uk/expression-of-interest)

#### **OCR Resources:** *the small print*

OCR's resources are provided to support the delivery of OCR qualifications, but in no way constitute an endorsed teaching method that is required by OCR. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

This resource may be freely copied and distributed, as long as the OCR logo and this small print remain intact and OCR is acknowledged as the originator of this work.

Our documents are updated over time. Whilst every effort is made to check all documents, there may be contradictions between published support and the specification, therefore please use the information on the latest specification at all times. Where changes are made to specifications these will be indicated within the document, there will be a new version number indicated, and a summary of the changes. If you do notice a discrepancy between the specification and a resource please contact us at: [resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk).

OCR acknowledges the use of the following content:

Front cover: Designing, baranq/Shutterstock.com; Square down and Square up: alexwhite/Shutterstock.com

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: [resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk)

#### **Looking for a resource?**

There is now a quick and easy search tool to help find **free** resources for your qualification:

[www.ocr.org.uk/i-want-to/find-resources/](http://www.ocr.org.uk/i-want-to/find-resources/)

[www.ocr.org.uk](http://www.ocr.org.uk)

OCR Customer Contact Centre

#### **Vocational qualifications**

Telephone 02476 851509

Facsimile 02476 851633

Email [vocational.qualifications@ocr.org.uk](mailto:vocational.qualifications@ocr.org.uk)

OCR is part of Cambridge Assessment, a department of the University of Cambridge. *For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored.*

© **OCR 2018** Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England. Registered office The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA. Registered company number 3484466. OCR is an exempt charity.



**Cambridge  
Assessment**

