LEVEL 2

WJEC Level 2 Additional Mathematics

Approved by Qualifications Wales

Guidance for Teaching: Unit 6

Teaching from 2026
For award from 2027

This Qualifications Wales regulated qualification is not available to centres in England.

Made for Wales.
Ready for the world.

WJEC would like to thank the students and teachers of Fitzalan High School, Cardiff and Llanwern High School, Newport for their participation in the creation of these covers.

Contents

© WJEC CBAC Ltd. 2026

Introduction

WJEC Level 2 Additional Mathematics has been approved by Qualifications Wales and is available to all centres in Wales. It will be awarded for the first time in Summer 2027, using grades Pass, Merit or Distinction.

Aims of the Guidance for Teaching

The principal aim of the Guidance for Teaching is to support teachers in the delivery of WJEC Level 2 Additional Mathematics and to offer guidance on the requirements of the qualification and the assessment process. The Guidance for Teaching is **not intended as a comprehensive reference**, but as support for teachers to develop stimulating and exciting courses, tailored to the needs and skills of their learners. The guide offers possible classroom activities and links to useful resources (including our own, freely available digital materials and some from external sources) to provide ideas for immersive and engaging lessons.

Additional ways that WJEC can offer support:

- sample assessment materials and mark schemes
- professional learning events
- examiners' reports on each unit
- direct access to the subject officer
- free online resources
- Exam Results Analysis
- Online Examination Review.

© WJEC CBAC Ltd. 2026

Qualification Structure

WJEC Level 2 Additional Mathematics consists of six units (two mandatory, four optional). The qualification is unitised and does not contain tiering. There is no hierarchy to the order the units should be taught.

	Unit title	Type of Assessment	Weighting
Mandatory Units			
Unit 1	Algebra	Written examination	33⅓%
Unit 2	Calculus	Written examination	33⅓%
Optional Units			
Unit 3	Geometry and Trigonometry	Written examination	33⅓%
Unit 4	Statistics	Written examination	33⅓%
Unit 5	Mechanics	Written examination	33⅓%
Unit 6	Discrete and Decision Mathematics	Written examination	33⅓%

To be awarded the qualification, learners must complete **three** units:

- **two** mandatory units
- **one** optional unit.

Learners who complete fewer than three units will receive unit certification for the successful completion of each unit.

© WJEC CBAC Ltd. 2026

Unit 6 Summary of Assessment

Unit 6: Discrete and Decision Mathematics **Written examination: 50 minutes** **33⅓% of qualification** **40 marks**
The paper will comprise a number of short and longer, both structured and unstructured, questions. A calculator will be allowed in this paper.

Overview of Unit 6

Discrete and Decision Mathematics
(33⅓% of the qualification)

The purpose of this unit is to introduce and develop an understanding of new concepts and mathematical approaches relating to discrete and decision mathematics and be able to apply them to novel and abstract situations.

A calculator will be allowed in this examination.

In this unit, learners will develop knowledge, skills and understanding in:	
6.1	Graphs and networks terminology
6.2	Algorithms & algorithms on networks
6.3	Critical path analysis
6.4	Linear programming

Unit 6 Assessment objectives and weightings

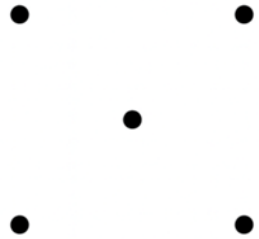
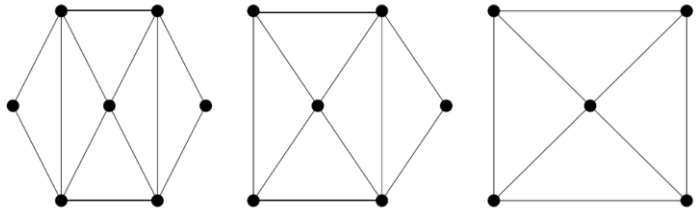
AO1	Recall and use their knowledge of the prescribed content.	18⅓%
AO2	Select and apply mathematical methods.	6⅔%
AO3	Interpret and analyse problems and use mathematical reasoning to solve them.	8⅓%

© WJEC CBAC Ltd. 2026

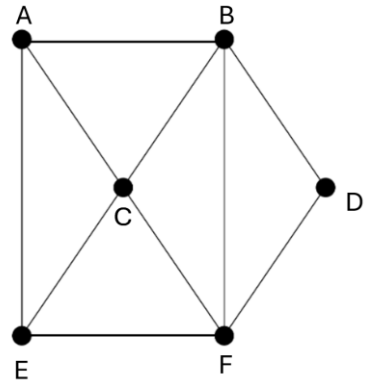
Unit 6 Teacher Guidance

6.1 Graphs and Networks		
	Content Amplification	**Teacher Guidance**
6.1.1 Graphs and networks terminology	Learners should know the meaning of the following terms and phrases in the context of graphs and networks: • edge • directed edge/arc • vertex/node • vertex degree • loop • graph/network • weighted graph/network • subgraph • simple graph • walk • connected graph • complete graph • open walk • closed walk • trail • path • cycle • tree • spanning tree • minimum spanning tree (MST). **See definitions in Appendix B of this specification.**	There are many equivalent definitions for several of the terms listed, learners are not expected to know them all. A glossary of acceptable definitions for Unit 6 can be found in Appendix B of the specification. WJEC Level 2 Additional Mathematics Specification-e **Example Questions** • Give an example of a connected graph. • Define the following graph theory terms: (i) Vertex (ii) Subgraph (iii) Path.

© WJEC CBAC Ltd. 2026

6.1.2 Eulerian and semi-Eulerian graphs	Learners should be able to: • state the degree of each of the vertices for a given graph • construct a graph with a given set of vertices, each with a specified degree. Learners should be able to justify whether a given graph is: • Eulerian • semi-Eulerian • neither. Learners should be aware of how adding appropriate edges can transform a non-Eulerian graph into a Eulerian graph, or a semi-Eulerian graph.	**Example Questions** • Using the vertices below, draw a simply connected graph with 4 vertices of degree 3 and 1 vertex of degree 4.  • Classify the following graphs as either Eulerian, Semi-Eulerian, or neither. Justify your answer: (i) (ii) (iii)  A 'non-Eulerian' graph is any graph that is not Eulerian. Therefore, a Semi-Eulerian graph is also non-Eulerian. **Example Question** (a) State the vertex degree of each of the vertices in the graph below. (b) State **one** edge that could be added or repeated to turn the graph below into a Eulerian graph and justify your answer.

© WJEC CBAC Ltd. 2026

		 Learners are **not expected** to apply the route inspection algorithm.

© WJEC CBAC Ltd. 2026

6.2 Algorithms & algorithms on networks

6.2.1 Algorithms and flow charts

Learners should know some of the properties of an algorithm, and why they are necessary, such as:

- finiteness
- definiteness
- correctness.

Learners should know the meaning of common flow chart symbols in the context of an algorithm. This might include:

- start/end
- arrow
- input/output
- process
- decision.

Learners should be able to interpret and apply simple algorithms represented as:

- a list of instructions
- a flow chart.

Example Question
Explain why the following is not a suitable algorithm to randomly order the digits from 1 to 20.

Step 1: Roll a fair 20-sided die.
Step 2: If the number has not previously been rolled, print the result of the roll and go to Step 3. If the number has previously been rolled, return to Step 1.
Step 3: If you have printed 20 numbers STOP. If you have not printed 20 numbers return to Step 1.

(The algorithm may not terminate after a finite number of steps, therefore does not adhere to the finiteness property).

The following flow chart symbols will be used:

Flowchart symbol	Name	Description
	Flow lines	Used to connect other symbols and show the direction of flow.
	Start/stop	Used to represent the start/stop of the flowchart.
	Input/Output	Represents information that is given as an input (initial values for variables) and output (printed values).

© WJEC CBAC Ltd. 2026

	Processing	Any process, for example, arithmetic operation "$c = a + b$"
	Decision	Used to check any condition where there are two possible answers Yes/True or No/false.

Example questions:

1. Algorithm represented as a list of instructions

The following algorithm can be used to find the highest common factor of two numbers a and b where $a < b$.

Step 1: Find the largest positive integer q such that $b = aq + r$ for some integer $r \geq 0$.
Step 2: If $r = 0$ print 'the HCF is a' and STOP. If $r > 0$ go to Step 3.
Step 3: Let $b = a$ and $a = r$ and return to Step 1.

Use the algorithm, and the table below to find the HCF of 12 and 45.

An example of the application of the algorithm might look like:

b	a	q	r	PRINT
45	12	3	9	
12	9	1	3	
9	3	3	0	The HCF is 3

Or it might look like:

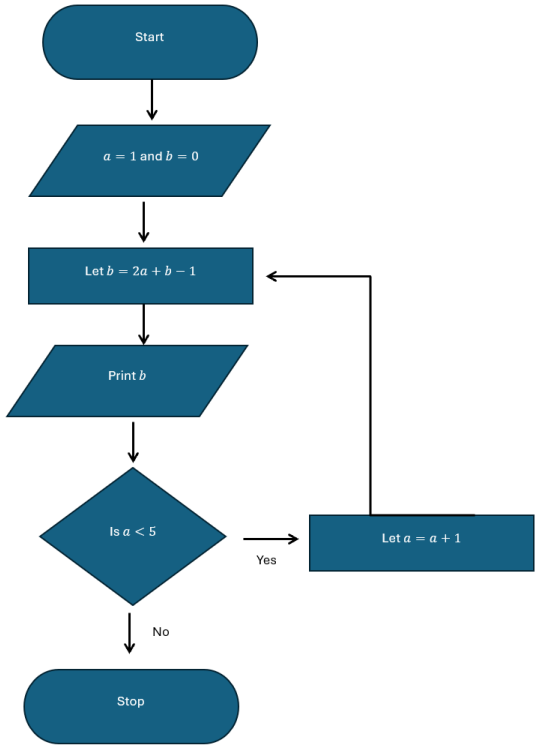
© WJEC CBAC Ltd. 2026

b	a	q	r	PRINT
45	12			
		3	9	
12	9			
		1	3	
9	3			
		3	0	The HCF is 3

2. Algorithm represented as a flow-chart

Using the table to apply the algorithm below, explain what the algorithm represented by the flow chart does:

© WJEC CBAC Ltd. 2026



An example of the algorithm being applied might look like:

a	b	PRINT
1	0	
	1	1
2	4	4
3	9	9

© WJEC CBAC Ltd. 2026

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr><td>4</td><td>16</td><td>16</td></tr>
<tr><td>5</td><td>25</td><td>25</td></tr>
</table>

The algorithm prints the first five square numbers, starting with 1.
</td>
</tr>
<tr>
<td>6.2.2
Sorting algorithms</td>
<td>Learners should be able to apply sorting algorithms to lists of distinct letters or distinct numbers, in either ascending or descending order. This includes using:
• Bubble sort
• Quick sort (pivot = middle, or middle right if two middle values are possible).</td>
<td>Bubble sort: Learners only need to indicate what the sorted list would look like after each pass, not after each comparison.

Example Question

Use the bubble sort to sort the following list of numbers into ascending order:
3 5 2 6 7 1 4

An example of a solution might look like:

Pass 1: 3 2 5 6 1 4 7
Pass 2: 2 3 5 1 4 6 7
Pass 3: 2 3 1 4 5 6 7
Pass 4: 2 1 3 4 5 6 7
Pass 5: 1 2 3 4 5 6 7
Pass 6: 1 2 3 4 5 6 7

No swaps were made in pass 6, therefore stop.

Note: numbers in bold need not be compared in subsequent passes.

Quick sort: Learners only need to indicate what the sorted list would look like after each pass, not after each comparison, and should also indicate the pivots chosen.

Example Question

Use the quick sort to sort the following list of numbers into</td>
</tr>
</table>

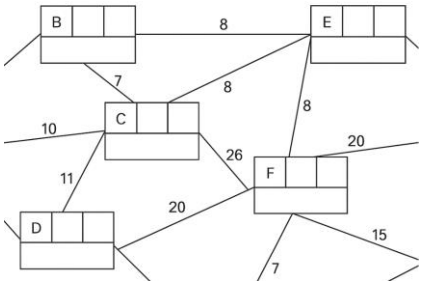
© WJEC CBAC Ltd. 2026

		ascending order: 3 5 2 6 7 1 4 An example of a solution might look like: Pivot(s) 3 5 2 **6** 7 1 4 | 6 Pass 1: 3 5 **2** 1 4 6 **7** | 2, (7) Pass 2: **1** 2 3 **5** 4 6 7 | (1), 5 Pass 3: 1 2 **3** 5 **4** 6 7 | (3), (4) All the items in the list have been chosen as pivots, therefore the list is ordered, stop. Note: it is not essential to choose pivots in sub lists of size 1 (indicated in brackets above). Named algorithms, such as the quick sort and bubble sort will be available as part of the “formula sheet” at the front of the examination paper, please see the SAMs for more details. Learners are not expected to know about the order of an algorithm.

© WJEC CBAC Ltd. 2026

6.2.3 Minimum spanning tree algorithms	Learners should be able to apply Kruskal's algorithm **or** Prim's algorithm to a given weighted network to find a minimum spanning tree (MST). Learners should know that the algorithm does not always lead to a unique solution.	Learners will be able to choose which algorithm to apply and will be required to name the algorithm. If a learner chooses Kruskal's algorithm, they must write the order in which the edges are chosen, and provide evidence of rejecting an edge, where appropriate. If a learner chooses Prim's algorithm, the learner must write the order in which the edges are chosen, and there should be no evidence of any edges being rejected. If a choice of edges is possible, learners are free to choose one correct edge. **Example Question** (a) Use an appropriate algorithm to find the MST for the following network. You must state the name of the algorithm that you use. (b) In addition to the MST found in (a) list the arcs of another MST. (c) Explain why the MST you found in (a) is not unique. Learners are not expected to use Prim's algorithm on a matrix. Learners may be given questions in context, for example, relating to physical networks, or infrastructure.

© WJEC CBAC Ltd. 2026

6.2.4 Dijkstra's algorithm	Learners should be able to apply Dijkstra's algorithm to a given weighted network, to find a path of minimum weight.	The network will be given with the necessary boxes for the order of labelling and working values. A key will be provided as a reminder so learners can consistently record their working values: **Key:** **Vertex** | **Order of labelling** | **Final value** **Working values** Part of a network and how it might be presented to learners:  Learners will not be expected to draw the network, nor the boxes around the vertices. Learners may be given questions in context, for example, finding a route of minimum length (of time, or distance) between two towns, via possible roads. Learners **will not** be expected to solve problems relating to future restrictions on the network. For example, "road AD is now closed, state the new path of minimum weight".

© WJEC CBAC Ltd. 2026

6.3 Critical path analysis

6.3.1 Activity on arc networks, precedence tables and dummies

Learners should be able to form a precedence table for a given list of precedence statements.

Learners should be able to model a project by constructing an activity on arc network from a given precedence table. The use of a dummy activity is not included.

Learners should know that dummy activities are placeholders and do not represent actual tasks, and, as such, do not take up time or resources.

Learners should know that a dummy activity may be used for a given activity on arc network to:

- ensure unique start and end nodes
- show a required precedence.

Learners are not expected to justify the use of a dummy, but they are expected to be able to solve problems on networks that make use of dummies.

Learners may be given questions in context, for example, the activities could relate to a set of instructions to complete a task.

The phrase “activity network” may be used as shorthand for an activity on arc network.

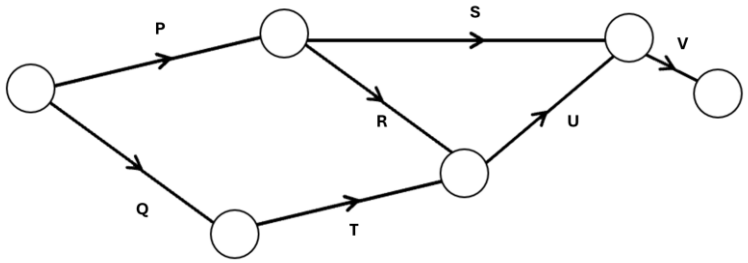
Learners will not be asked to produce activity on arc networks that require the use of a dummy activity.

Example Question

Using the precedence table below, construct an activity network:

Activity	Depends on
P	-
Q	-
R	P
S	P
T	Q
U	R, T
V	S, U

© WJEC CBAC Ltd. 2026



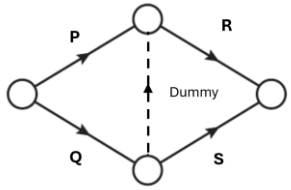
Note: Activities must be represented by the arcs.

Learners are not required to label the event nodes with numerical values, however, if they do, they should label the source node as 0, and subsequent events should follow the sequence 1, 2, 3, …

Dummies will be represented on activity networks by a dotted line.

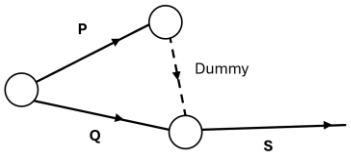
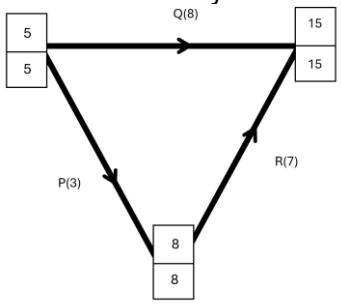
For example:

To show a required precedence

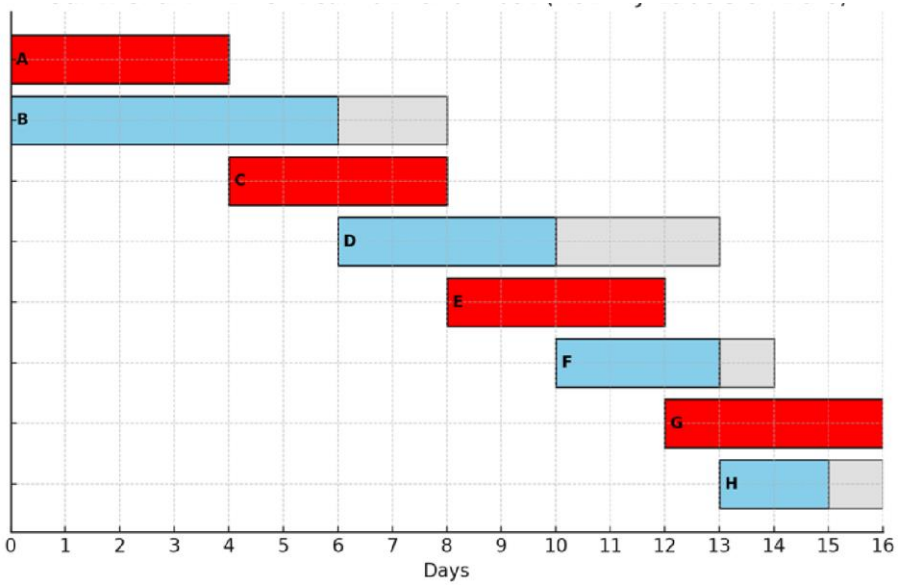


To ensure a unique start and end node

© WJEC CBAC Ltd. 2026

		
6.3.2 Critical paths	Learners should be able to calculate: • early event times • late event times for a given activity on arc network, by performing a forward/backward pass. Dummies may be used to ensure unique start and end nodes, and/or to show precedencies. Learners should be able to calculate the total float for activities and identify critical activities and critical paths from an activity on arc network, by considering the early and late event times.	The activity network will be given with the necessary boxes for the early and late event times. A key will be provided as a reminder so learners can consistently record their early and late event times. **Key:** **Early event times** **Late event times** **Example Question** Part of an activity network is shown below. List the critical activities: 

© WJEC CBAC Ltd. 2026

6.3.3 Gantt (cascade) charts	Learners should be able to interpret Gantt (cascade) charts. This includes: • identifying critical activities • critical paths • the earliest start time for an activity • the latest end time for an activity • the total float for an activity • identifying which activities must/may be happening at a given time. Note: time scales will indicate elapsed time.	Learners will not be expected to draw a Gantt (cascade) chart. Learners may be given a Gantt (cascade) chart for a particular relevant context. **Example Questions**  • Using the given Gantt chart, identify the activities that must be happening on day 7, and the activities that may be happening. • Using the given Gantt chart, list the critical activities. • What is the total float for activity B, according to the given Gantt chart?

© WJEC CBAC Ltd. 2026

6.4 Linear programming		
6.4.1 Linear programming terminology	Learners should know the meaning of the following terms: • decision variable • constraints • objective function • feasible solution • feasible region • optimal solution.	A glossary of acceptable definitions for Unit 6 can be found in Appendix B of the specification. WJEC Level 2 Additional Mathematics Specification-e
6.4.2 Setting up linear programming problems in two variables	Learners should be able to formulate simple constrained optimisation problems as linear programs, for two decision variables, for a given context. This could include identifying the appropriate objective function and stating whether it is to be maximised or minimised, and formulating appropriate inequalities for a given constraint. All decision variables will be subject to non-negativity constraints. Therefore, the feasibility region will be restricted to the first quadrant. Learners should be aware of the modelling cycle and how this applies to linear programming problems in terms of moving from the 'real life problem' to the associated 'mathematical problem' and the implication this has on the validity of the solutions.	Learners will be given questions in context. These will be restricted to simple cases, such as: • maximising profit when producing two types of toys • minimising cost when employing two types of worker. For example, we might be solving problems for discrete decision variables but are modelling the feasible region using lines that represent continuous decision variables. Possible other considerations could be: • Are all resources fully consumed or are there hidden costs or inefficiencies? • Is profit per unit fixed, or does it vary with the number produced (economies of scale)? • Are there other constraints (e.g. demand limits) not included?

© WJEC CBAC Ltd. 2026

6.4.3 Solving linear programming problems in two variables	Learners should be able to represent constraints and an objective line graphically. Learners should be able to use shading (to eliminate) and identify the feasibility region for a set of given or derived constraints. Learners should be able to solve linear programming problems involving two decision variables. Learners should use either the objective line method **or** the vertex testing method, to find a unique solution, including cases where integer solutions are required. In such cases, the optimal solution will be found at a point with integer coordinates. Learners are not expected to have any knowledge of the simplex algorithm for solving linear programming problems.	Linear constraints will be of the form: • $ax + by \geq c$ • $ax + by \leq c$ • $ay \geq bx$ • $ay \leq bx$ where, a, b, c are integers. **Example Question** Identify with a letter **R** the region that satisfies all five inequalities. Questions will be phrased so that learners will be able to choose to use either method. If using the objective line method, the optimum solution will either be the first (if minimising) or last (if maximising) intersection point between two lines, representing the constraints that are used to define the feasible region. Evidence of an objective line being drawn and used will be required. If using vertex testing, the optimum solution will be one of the finite points of intersection, and all (non-trivial) points of intersection will need to be tested. If the problem requires integer values for the decision variables, the optimum solution found by either method will be an integer coordinate. Learners will not be required to test around a non-integer coordinate to find the optimum integer coordinate inside the feasible region.

© WJEC CBAC Ltd. 2026

Learning Experiences

Learners should be encouraged to consider the following learning experiences and skills to further develop their understanding, appreciation and awareness of the subject content. Information in the table below provides opportunities for teachers to integrate the learning experiences into delivery.

Learning Experience	Exemplification of Learning Experience
Work both independently and collaboratively.	Much of the specification requires learners to develop their own skills independently. However, collaboration can be achieved through research and extension. For example, learners could be encouraged to conduct collaborative research projects exploring other common, and simple algorithms, such as for: • sorting • searching • packing.
Gain experience and appreciation of the role mathematics plays in other subjects and areas of the curriculum.	Section 6.2 is heavily linked with concepts explored in GCSE Computer Science, such as algorithms. Learners could use an appropriate coding language to implement some of the simple algorithms introduced in this section of the specification. Section 6.3 is linked with concepts explored in A-level Business, such as decision-making models. Section 6.4 is an application of many of the algebra and graphical skills explored in GCSE Mathematics and Numeracy. Indeed, students will define linear constraints using linear inequalities, feasible regions, by plotting lines and applying shading techniques similar to those explored in GCSE Mathematics and Numeracy. However, this will culminate in a meaningful and applicable solution that has “real-world” implications.
Gain awareness and appreciation of some of the different careers and work-related areas that draw upon mathematics.	Some examples of careers / roles that make use of aspects of the mathematics explored in this unit include: **Graphs and networks** • Transport planner / logistics analyst (designing and optimising road, rail, and flight networks) • Telecommunications network engineer (designing cable layouts) • Operations researcher (applying graph theory to scheduling and resource allocation problems) • Social network analyst (mapping influence or

© WJEC CBAC Ltd. 2026

	information spread) **Algorithms & Algorithms on Networks** • Supply chain analyst (minimising delivery costs or travel times) • Software engineer (optimisation algorithms) • Airline operations analyst (optimising aircraft routing) • Cyber security analyst (analysing network vulnerabilities) • Game developer (programming NPC movements using graph search algorithms) **Critical Path Analysis** • Construction project manager (scheduling build phases and subcontractor work) • Event coordinator (planning complex events with dependent tasks) • Production manager (scheduling film shoots, actors, post-production tasks) • Manufacturing operations planner (sequences processes to meet deadlines efficiently) • IT systems deployment manager (rolling out large-scale upgrades) **Linear programming** • Financial analyst (portfolio optimisation) • Agricultural planner (deciding drop allocations to maximise yield, or minimise environmental impact) • Energy grid optimiser (balancing supply and demand)
Access rich tasks that invoke curiosity, build resilience and require learners to be resourceful.	Such tasks should go beyond typical textbook questions. Examples of such tasks are: • Eulerian trial puzzles (in context). This could start with the classic “seven bridges” problem and extend to local attractions. • Network shortest route challenges. Give a transport network with changing weights (e.g. delays due to roadworks) and learners can adapt Dijkstra’s algorithm to find a new optimum route. • Scheduling with constraints (in context). Give a festival line-up with multiple stages and shared crew. Learners can then produce a feasible Gantt chart that meets all the constraints and then try to optimise to shorten the total completion time.

© WJEC CBAC Ltd. 2026

	• Optimising with a “twist”. Give learners a linear programming problem where the apparent optimum is invalid for a real-world reason. This could include a fraction of an item being produced or could include a “banned” combination. Some of the ideas suggested here go slightly beyond the specification, which could help build resilience and resourcefulness.
Undertake practical work that allows learners to apply their mathematical skills inside and outside of the classroom setting.	Learners can be given the opportunity to bring decision mathematics into a physical setting. Some examples of this include: • Network cabling simulation. Use strings and pins on a large board to physically construct a network between points (rooms, PCs, or devices) and then apply Prim’s/Kruskal’s algorithm to find a MST. • DIY project scheduling. In small groups, learners plan a simple build (e.g. making a display board, or organise the setup of an event). • Pop-up business optimisation. Learners simulate running a market stall with limited stock, budget, and time. They can then set up and solve a linear programming model based on the constraints, to maximise their profit, and then test this plan in a roleplay scenario.
Encounter familiar, unfamiliar and complex problems.	Learners will encounter a range of problems across the unit. Familiar problems could include: • determining whether a given network is Eulerian or semi-Eulerian • applying bubble sort to a short list • finding a minimum spanning tree from a small, weighted network • solving a linear programming problem in two variables using the objective line method. Unfamiliar problems might involve: • interpreting a critical path diagram in an unfamiliar context • applying Dijkstra’s algorithm to a network where the weights represent non-standard measures such as risk or energy usage • reasoning about the validity of a linear programming solution when decision variables must be integers. Complex problems could require:

© WJEC CBAC Ltd. 2026

	• combining multiple techniques, such as using a sorting algorithm to order edges before applying Kruskal's algorithm • modelling a project from a precedence table before identifying the critical path and representing the results as a Gantt chart • solving an optimisation problem that begins with formulating constraints from a real-world context, graphing them, and interpreting the feasible region before identifying the optimal solution.

© WJEC CBAC Ltd. 2026

Opportunities for embedding elements of the Curriculum for Wales

<table>
<tr><th colspan="4">Curriculum for Wales Strands</th></tr>
<tr><th colspan="4">Cross-cutting Themes</th></tr>
<tr><td rowspan="3">Local, National & International Contexts</td><td colspan="3">There are many opportunities to include Local, National and International Contexts in Level 2 Additional Mathematics. These opportunities are important to learners because they show them how mathematics applies to real situations, from local projects to global operations. It helps learners appreciate its relevance, transfer their skills, and understand how modelling and optimisation address challenges in areas such as transport, communication, construction, resource management, and logistics.

Below are some examples of how Local, National & International Contexts can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.2.4</td><td>Dijkstra’s algorithm</td><td>Learners could apply shortest-path problems to real maps of local towns, Welsh transport links, or international shipping routes to show direct relevance to transport planning.

Learners could model the quickest walking route between landmarks in Cardiff (or their local area), adjusting for pedestrian-only streets.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

<table>
<tr><td></td><td>6.3.2</td><td>Critical Path Analysis</td><td>Learners should be encouraged to use local events, such as organising a school concert, and international examples, such as Olympic venue construction schedules, to demonstrate how CPA applies across scales.

Learners could create a precedence table and Gantt chart for their school Eisteddfod set-up process.</td></tr>
<tr><td rowspan="3">Sustainability</td><td colspan="3">There are many opportunities to include Sustainability in Level 2 Additional Mathematics. These opportunities are important to Learners because they show how optimisation and efficient design can minimise waste, reduce energy use, and support environmentally responsible decision-making in real-world contexts.

Below are some examples of how Sustainability can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.2.3</td><td>Minimum Spanning Tree algorithms</td><td>Learners could show how network design can reduce environmental impact by minimising materials or energy usage.

Learners could use Prim’s algorithm to plan a minimum-length water pipe network for a rural community, reducing cost and environmental footprint.</td></tr>
<tr><td></td><td>6.4</td><td>Linear Programming</td><td>Learners could model production schedules that minimise waste or energy usage within given resource constraints.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

<table>
<tr><td></td><td></td><td></td><td>Learners could optimise the mix of crops grown on local farmland to meet certain demands with minimal fertiliser use.</td></tr>
<tr><td rowspan="3">Careers and Work-Related Experiences</td><td colspan="3">There are many opportunities to include Career and Work-Related Experiences (CWRE) in Level 2 Additional Mathematics. These opportunities are important to Learners because they highlight the relevance of decision mathematics to fields such as logistics, engineering, project management, IT, and business operations.

Below are some examples of how CWRE can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.1</td><td>Graphs and Networks</td><td>Learners can be given opportunities to link concepts to roles in logistics, IT network engineering, and operations research.

Teachers could invite a transport planner to demonstrate how network optimisation is used in their work.</td></tr>
<tr><td></td><td>6.3.2</td><td>Critical Path Analysis</td><td>Teachers could relate CPA to project management in construction, events, and manufacturing.

Learners could simulate scheduling for a film shoot, using CPA to avoid costly delays.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

<table>
<tr><th colspan="4">Cross-curricular Skills – Literacy</th></tr>
<tr><td rowspan="3">Listening</td><td colspan="3">There are many opportunities to include Literacy in Level 2 Additional Mathematics. These opportunities are important to Learners because they develop the ability to interpret written and diagrammatic information, articulate reasoning clearly, and communicate mathematical ideas effectively in spoken and written form.

Below are some examples of how Literacy can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.2</td><td>Algorithms and Algorithms on networks</td><td>Learners could be given oral instructions for algorithms. Working in pairs, one learner describes the steps of Kruskal’s algorithm while the other applies them to a graph.</td></tr>
<tr><td rowspan="3">Reading</td><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.2.1</td><td>Algorithms and flow charts</td><td>Learners can interpret written and diagrammatic algorithm instructions. Learners could read and execute a flow chart for bubble sort without verbal explanation.</td></tr>
<tr><td>6.4</td><td>Linear programming</td><td>Learners can interpret written real-world question and constraints. Learners could read a business scenario and identify the decision variables, objective function, and constraints before modelling the problem graphically.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

Speaking	***Specification Reference***	***Amplification***	***Example***
	6.2.3	Minimum spanning tree algorithms	Learners can explain the reasoning behind their choice of edges when applying Prim's or Kruskal's algorithm. Learners could then present to the class why a particular sequence of edges was chosen to produce the minimum spanning tree.
	6.4.2	Setting up linear programming problems	Learners can be given opportunities to explain their reasoning for modelling constraints in context. Learners could explain to a partner how each constraint in a linear programming model reflects a real-world limitation in the given scenario.
Writing	***Specification Reference***	***Amplification***	***Example***
	6.3.2	Identifying critical paths	Learners could produce clear written solutions showing each stage of the forward and backward pass. Learners could write a step-by-step solution for calculating early and late event times, identifying the critical path.
	6.4.3	Solving linear programming problems	Learners can communicate solutions clearly using written explanations alongside graphs. Learners could write a full solution to a linear programming problem, including graph, feasible region, and justification of the optimal solution.

© WJEC CBAC Ltd. 2026

<table>
<tr><th colspan="4">Cross-curricular Skills – Numeracy</th></tr>
<tr><td></td><td colspan="3">There are many opportunities to include Numeracy in Level 2 Additional Mathematics. These opportunities are important to Learners because they will apply arithmetic, algebra, geometry, and graphical skills to meaningful problems, reinforcing and extending core mathematical proficiency.

Below are some examples of how Numeracy can be embedded into teaching and learning:</td></tr>
<tr><td rowspan="3">Developing Mathematical Proficiency</td><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>Throughout Unit 6</td><td></td><td>Learners can apply arithmetic, algebra, and graphical reasoning in varied decision contexts.
Learners could calculate early / late event times, or total float in critical path analysis problems, accurately, and consistently.</td></tr>
<tr><td>6.2</td><td>Algorithms & algorithms on networks</td><td>Learners should be given opportunities to become proficient at choosing and applying the correct algorithm to solve a given problem.</td></tr>
<tr><td rowspan="3">Understanding the number system helps us to represent and compare relationships between numbers and quantities</td><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.2.4</td><td>Dijkstra’s algorithm</td><td>Learners can compare path lengths and interpret weighted edges as numerical relationships.

Learners could determine which delivery route minimises fuel use when given distance data.</td></tr>
<tr><td>6.2.2</td><td>Sorting algorithms</td><td>Learners should be given opportunities to apply the bubble and quick sort algorithms in a range of contexts.

Learners will need to compare numbers and quantities when implementing sorting algorithms.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

	Specification Reference	***Amplification***	***Example***
Learning about geometry helps us understand shape, space and position and learning about measurement helps us quantify in the real world	6.1	Graphs and Networks	Learners should be given opportunities to interpret and construct accurate network diagrams from spatial layouts, ensuring vertices and edges reflect real-world positions and distances. Learners could draw a network map of a school site showing buildings as vertices and paths as weighted edges, then use it for a shortest-route problem.
	6.3.2	Critical Path Analysis	Using diagrammatic representations of networks to understand spatial layout and sequence. Translating a site plan into an activity-on-arc network.

© WJEC CBAC Ltd. 2026

<table>
<tr><th colspan="4">Cross-curricular Skills – Digital Competence</th></tr>
<tr><td rowspan="3">Producing</td><td colspan="3">There are many opportunities to include Digital Competence in Level 2 Additional Mathematics. These opportunities are important to Learners because they enable the use of digital tools to represent, analyse, and solve problems, and develop computational thinking through algorithmic processes.

Below are some examples of how Digital Competence can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.3.3</td><td>Gantt charts</td><td>Learners should be given opportunities to use spreadsheet for project management software to generate outputs from CPA data.

Learners could produce a Gantt chart in Excel or MS Project for a school production schedule.</td></tr>
<tr><td></td><td>6.4.3</td><td>Solving linear programming problems</td><td>Learners should be given opportunities to use graphing software or dynamic geometry tools to plot constraints, identify the feasible region, and determine the optimal solution.

Learners could use Desmos or GeoGebra to graph inequalities and visually identify the optimal vertex in a linear programming problem.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

	Specification Reference	***Amplification***	***Example***
Data and Computational Thinking	6.2	Algorithms on networks	Learners should have opportunities to apply logical sequencing and debugging skills when implementing algorithms digitally. Learners could code Dijkstra's algorithm in Python and test on sample data.
	6.2.2	Sorting algorithms	Learners should have opportunities to compare algorithm efficiency by coding and testing different sorting algorithms on varied datasets. Learners could Implement bubble sort and quick sort in Python, timing each on random lists, and analyse the performance difference.

© WJEC CBAC Ltd. 2026

<table>
<tr><th colspan="4">Integral Skills</th></tr>
<tr><td rowspan="3">Creativity and Innovation</td><td colspan="3">There are many opportunities to include Creativity and Innovation in Level 2 Additional Mathematics. These opportunities are important to Learners because they encourage the design of novel problem contexts and the exploration of alternative methods to reach valid solutions.

Below are some examples of how Creativity and Innovation can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.4</td><td>Linear Programming</td><td>Learners should be given opportunities to design novel contexts for optimisation problems.

Learners could invent a fictional festival and create linear programming constraints to maximise ticket sales.</td></tr>
<tr><td></td><td>6.1</td><td>Graphs and networks</td><td>Teachers should encourage learners to design their own network problems from real-world layouts, choosing suitable weights and contexts.

Learners could create a transport network for a fictional city and propose challenges such as finding an MST or shortest route.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

<table>
<tr><td rowspan="3">Critical Thinking and Problem Solving</td><td colspan="3">There are many opportunities to include Critical Thinking and Problem Solving in Level 2 Additional Mathematics. These opportunities are important to Learners because they require the analysis of complex problems, the evaluation of different strategies, and the justification of decisions.

Below are some examples of how Critical Thinking and Problem Solving can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.1.2</td><td>Eulerian and semi-Eulerian graphs</td><td>Learners should be given opportunities to justify classification decisions and adapt networks to meet criteria.

Learners could modify a delivery route network to become semi-Eulerian and explain the change.</td></tr>
<tr><td></td><td>6.4</td><td>Linear programming</td><td>Learners should be given opportunities to justify their choice of constraints and interpret the impact of changes to the model.

Learners could modify constraints in a linear programming problem to reflect a supply shortage and explain how the solution changes.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

<table>
<tr><td rowspan="3">Planning and Organisation</td><td colspan="3">There are many opportunities to include Planning and Organisation in Level 2 Additional Mathematics. These opportunities are important to Learners because they involve structuring tasks, managing resources, and sequencing activities to achieve efficient and timely outcomes.

Below are some examples of how Planning and Organisation can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>6.3</td><td>Critical Path Analysis</td><td>Learners should be given opportunities to structure multi-stage projects with clear timelines and dependencies.

Learners could plan the build of an exhibition stand using CPA and ensuring resource allocation meets deadlines.</td></tr>
<tr><td></td><td>6.3.1</td><td>Activity-on-arc networks</td><td>Learners should be given opportunities to plan project stages logically, ensuring precedence is respected and resources are used efficiently. Learners could sequence activities for a school event so all set-up tasks finish in time for the opening.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

<table>
<tr><td rowspan="3">Personal Effectiveness</td><td colspan="3">There are many opportunities to include Personal Effectiveness in Level 2 Additional Mathematics. These opportunities are important to Learners because they develop resilience, adaptability, and self-management when tackling challenging tasks and monitoring progress towards a solution.

Below are some examples of how Personal Effectiveness can be embedded into teaching and learning:</td></tr>
<tr><td>Specification Reference</td><td>Amplification</td><td>Example</td></tr>
<tr><td>All sections, with emphasis on 6.3 Critical Path Analysis and 6.4 Linear Programming</td><td></td><td>Learners should be encouraged to take ownership of their problem-solving process, manage their time effectively when working on multi-stage decision mathematics problems, and adapt strategies when encountering obstacles.

Learners could be given a real-world scheduling or optimisation problem with incomplete information. Learners would need to identify what additional data is needed, plan a strategy to obtain and use it, monitor their progress against the problem’s constraints, and reflect on how effectively they managed their approach.</td></tr>
<tr><td>6.2.4</td><td>Dijkstra’s algorithm</td><td>Learners should be encouraged to build resilience by tackling multi-step problems that require checking, revising, and verifying solutions without prompts.

Learners could solve a shortest-path problem, then verify the solution by manually checking all alternative routes for errors.</td></tr>
</table>

© WJEC CBAC Ltd. 2026

Glossary for Unit 6

6.1: Graphs and Networks	
Term	**Definition**
Edge	An edge is a line segment that connects two vertices in a graph.
Directed edge	An edge that has an orientation and can only be traversed in one direction. Also known as an arc.
Vertex/node	A vertex is a point on a graph that may or may not be connected to other vertices by edges. Note: Vertices are also known as points.
Vertex degree	The number of edges that are **incident** to the vertex. A vertex is even/odd if the degree is even/odd. Note: vertex degree is also known as valency, or the “order” of a vertex.
Loop	An edge that starts and finishes at the same vertex.
Graph/network	A graph G consists of vertices that are connected by edges. Note: Graphs are also known as networks in certain contexts.
Weighted graph/network	A graph/network where each edge has a numerical value (weight) assigned to it.
Subgraph	A graph G’ whose vertices and edges belong to G.
Simple graph	A graph in which there are no loops, no directions on the edges, and where there is at most one edge connecting any pair of vertices.
Walk	A route through a graph along edges from one vertex to the next.
Connected graph	A graph in which every vertex is connected by a walk.
Complete graph	A graph in which every vertex is connected to every other vertex by a unique edge.
Open walk	A walk where the start vertex is not the same as the end vertex.
Closed walk	A walk where the start vertex is the same as the end vertex.
Trail	A walk in which no edge is visited more than once.
Path	A walk in which no vertex is visited more than once.
Cycle	A walk in which the end vertex is the same as the start vertex and no other vertex is visited more than once.
Tree	A connected graph with no cycles.

© WJEC CBAC Ltd. 2026

Spanning tree	A subgraph of G that includes all vertices and is also a tree.
Minimum spanning tree (MST)	A spanning tree such that the total weight of its edges is as small as possible. Note: an MST is also known as a minimum connector.
Eulerian graph	A connected graph where every vertex has an even degree.
Semi-Eulerian graph	A connected graph where precisely two vertices have an odd degree.
6.2: Algorithms & algorithms on networks	
Term	**Definition**
Finiteness	The algorithm terminates after a finite number of steps.
Definiteness	Each step in the algorithm is precisely defined and unambiguous.
Correctness	The algorithm solves the given problem without errors or mistakes.

© WJEC CBAC Ltd. 2026

6.3: Critical path analysis	
Term	**Definition**
Precedence/dependence table	A table that shows the activities that must be completed before others are started.
Activity on arc network	Activities are represented by arcs, and nodes represent the start and end of activities.
Duration	The length of time it takes to complete an activity.
Critical activity	Any activity in which an increase in its duration would result in a corresponding increase in the duration of the whole project.
Critical path	A path from the start/source vertex/node to the end/sink vertex/node that entirely follows critical activities.
Early event time	The earliest time of arrival at the event allowing for the completion of all preceding activities.
Late event time	The latest time that an event can be left without extending the time needed for the project.
Total float	The amount of time that its start may be delayed without affecting the duration of the project.
Gantt (cascade) chart	A graphical way to represent the range of possible start and finish times for all activities on a single diagram.
6.4: Linear programming	
Term	**Definition**
Decision variable	A variable associated with linear programming problems. A solution is formed by assigning appropriate values to all decision variables.
Constraint	A restriction on the values of the decision variables.
Objective function	The function that defines the criteria for evaluating the solution.
Feasible solution	Values for the decision variables that satisfy each of the constraints in the linear programming problem.
Feasible region	The region that contains all the feasible solutions.
Optimal solution	A feasible solution that has the optimum objective function value.

© WJEC CBAC Ltd. 2026