CAMBRIDGE ADVANCED NATIONALS

OCR
Oxford Cambridge and RSA

Specification

LEVEL 3 ALTERNATIVE ACADEMIC QUALIFICATION
CAMBRIDGE ADVANCED NATIONAL IN

COMPUTING: APPLICATION DEVELOPMENT

Certificate H029
Extended Certificate H129

For first teaching in 2025

Version 2.0 (June 2024)

Tell us what you think

Your feedback plays an important role in how we develop, market, support and resource qualifications now and into the future. Here at OCR, we want teachers and students to enjoy and get the best out of our qualifications and resources, but to do that we need honest opinions to tell us whether we're on the right track or not. That's where you come in.

You can email your thoughts to ProductDevelopment@OCR.org.uk or visit the OCR feedback page to learn more about how you can help us improve our qualifications.

Designing and testing in collaboration with teachers and students

Helping young people develop an ethical view of the world

Equality, diversity, inclusion and belonging (EDIB) are part of everything we do

Are you using the latest version of this specification?

The latest version of our specifications will always be on **our website** and may differ from printed versions. We will inform centres about changes to specifications.

Disclaimer

Specifications are updated over time. Whilst every effort is made to check all documents, there may be contradictions between published resources 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**

Copyright

Contents

1 Why choose OCR?

Choose OCR and you've got the reassurance that you're working with one of the UK's leading exam boards. We've developed our specifications in consultation with teachers, employers, subject experts and higher education institutions (HEIs) to give students a qualification that's relevant to them and meets their needs.

We're part of Cambridge University Press & Assessment. We help millions of people worldwide unlock their potential. Our qualifications, assessments, academic publications and original research spread knowledge, spark curiosity and aid understanding around the world.

We work with a range of education providers in both the public and private sectors. These include schools, colleges, HEIs and other workplaces. Over 13,000 centres choose our A Levels, GCSEs and vocational qualifications including Cambridge Nationals and legacy Cambridge Technicals.

1.1 Our specifications

We provide specifications that help you bring the subject to life and inspire your students to achieve more.

We've created teacher-friendly specifications based on extensive research and engagement with the teaching community. Our specifications are designed to be straightforward to deliver and accessible for students. The design allows you to tailor the delivery of the course to suit your needs.

1.2 Our support

We provide a range of support services to help you at every stage, from preparation to delivery:

- A wide range of high-quality creative resources including resources created by leading organisations in the industry.
- Textbooks and teaching and learning resources from leading publishers. The Cambridge Advanced Nationals page on our website has more information about all the published support for the qualifications that we have endorsed.
- Professional development for teachers to meet a range of needs. To join our training (either face-to-face or online) or to search for training materials, go to the **Professional Development page** on our website.
- **Active Results** which is our free results analysis service. It helps you review the performance of individual students or whole groups.
- **ExamBuilder** which is our free question-building platform. It helps you to build your own tests using past OCR exam questions.
- OCR Subject Advisors, who give information and support to centres. They can help with specification and non examined assessment (NEA) advice, updates on resources developments and a range of training opportunities. They use networks to work with subject communities and share ideas and expertise to support teachers.

1.2.1 More help and support

Whether you are new to OCR or already teaching with us, you can find useful information, help and support on our **website**. Or get in touch:

support@ocr.org.uk
@ocrexams
01223 553998

1.3 Aims and learning outcomes

Our Cambridge Advanced Nationals in Computing: Application Development will encourage students to:

- develop key knowledge, understanding and skills, relevant to the subject
- think creatively, innovatively, analytically, logically and critically
- develop valuable communication skills that are important in all aspects of further study and life
- develop transferable learning and skills, such as communication, creativity, critical thinking, independent learning, problem solving and time management, that are important for progression to HE and can be applied to real-life contexts and work situations
- develop independence and confidence in applying the knowledge and skills that are vital for progression to HE and relevant to the ICT practitioners sector and more widely.

1.4 What are the key features of this specification?

The key features of OCR's Cambridge Advanced Nationals in Computing: Application Development for you and your students are:

- a simple and intuitive assessment model, that has:
    - externally assessed units, which focus on subject knowledge and understanding
    - applied and practical non examined assessment units (NEA)
    - optional NEA units to provide flexibility
- a specification developed with teachers specifically for teachers. The specification lays out the subject content, assessment criteria, teacher guidance and delivery requirements clearly
- a flexible support package made based on teachers' needs. The support package will help teachers to easily understand the qualification and how it is assessed
- a team of OCR Subject Advisors who directly support teachers
- a specification designed to:
    - complement A Levels in a Post-16 curriculum
    - develop wider transferable skills, knowledge and understanding desired by HEIs. More detail about the transferable skills these qualifications may develop is in **Section 5.3**.

All Cambridge Advanced National qualifications offered by OCR are regulated by Ofqual, the Regulator for qualifications offered in England.

The qualification numbers for OCR's Alternative Academic Qualification Cambridge Advanced Nationals in Computing: Application Development are:

- Certificate: QN 610/3974/3
- Extended Certificate: QN 610/3975/5

1.5 Acknowledgements

We would like to acknowledge the following Higher Education Providers for their input and support in designing these qualifications:
Anglia Ruskin University
Bournemouth University
University of Gloucestershire
University of Liverpool
University of Manchester
University of Staffordshire
University of the West of England
University of Westminster

2 Qualification overview

2.1 OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate) at a glance

Qualification number	610/3974/3
First entry date	01 September 2025
Guided learning hours (GLH)	150
Total qualification time (TQT)	200
OCR entry code	H029
Approved age range	16-18, 18+, 19+
Offered in	England only
Performance table information	This qualification is designed to meet the Department for Education's requirements for qualifications in the Alternative Academic Qualifications category of the 16-19 performance tables.
Eligibility for funding	This qualification meets funding approval criteria.
UCAS Points	This qualification is recognised in the UCAS tariff tables. You'll find more information on the **UCAS website**.
This qualification is suitable for students who:	• are age 16-19 and on a full-time study programme • want to develop applied knowledge and skills in application development • want to progress onto other related study, such as higher education courses in Computer Science, Computing and User Experience Design.
Entry requirements	There is no requirement for students to achieve any specific qualifications before taking this qualification.
Qualification requirements	Students must complete two units: • one externally assessed unit • one NEA unit
Assessment method/model	Unit F160 is assessed by an exam and marked by us. You will assess the NEA unit and we will moderate it. The NEA assignments are live for 2 years. The front cover details the intended cohort. You must make sure you use the live assignment that relates to the student's cohort for assessment and submit in the period in which the assignments are live. For example, a cohort beginning a 2-year course in September 2026 should use the set of assignments marked as being for 2026-2028 so that whatever order assignments are taken in, they will be able to re-submit improved work on the same NEA assignment if they wish to during their study of the qualification.

	Centres should avoid allowing new cohorts to use assignments which have already been live for a year, e.g. students who start the course in September 2027 using assignments for the 2026-2028 cohorts. Centres must have suitable controls in place to ensure that NEA assignment work is completed by each student independently and must not allow previously completed work for assignments which are still live to be shared as examples with other students.
Exam series each year	• January • June
Exam resits	Students can resit the examined unit twice before they complete the qualification.
NEA submission	There are two windows each year to submit NEA outcomes and request a moderation visit by an OCR Assessor. You must make unit entries for students before you can submit outcomes to request a visit. All dates are on our administration pages.
Resubmission of students' NEA work	If students have not performed at their best in the NEA Assignments, they can improve their work and submit it to you again for assessment. They must have your agreement and you must be sure it is in the student's best interests. We use the term 'resubmission' when referring to student work that has previously been submitted to OCR for moderation. Following OCR moderation, a student can attempt to improve their work for you to assess and provide the final mark to us. There is one resubmission opportunity per NEA assignment. All work submitted (or resubmitted) must be based on the assignment that is live for assessment. For information about feedback see **Section 6**. The final piece of work must be completed solely by the student and teachers must not detail specifically what amendments should be made.
Grading	Information about unit and qualification grading is in **Section 5.**

2.2 OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate) at a glance

Qualification number	610/3975/5
First entry date	01 September 2025
Guided learning hours (GLH)	360
Total qualification time (TQT)	500
OCR entry code	H129
Approved age range	16-18, 18+, 19+
Offered in	England only
Performance table information	This qualification is designed to meet the Department for Education's requirements for qualifications in the Alternative Academic Qualifications category of the 16-19 performance tables.
Eligibility for funding	This qualification meets funding approval criteria.
UCAS Points	This qualification is recognised in the UCAS tariff tables. You'll find more information on the **UCAS website**.
This qualification is suitable for students who:	• are age 16-19 and on a full-time study programme • want to develop applied knowledge and skills in application development • want to progress onto other related study, such as higher education courses in Computer Science, Computer Games Development, Computing, Creative Computing, Web and Mobile Development, Web and User Experience Design.
Entry requirements	There is no requirement for students to achieve any specific qualifications before taking this qualification.
Qualification requirements	Students must complete five units: • two externally assessed units • three NEA units
Assessment method/model	Units F160 and F161 are assessed by an exam and marked by us. You will assess the NEA units and we will moderate them. The NEA assignments are live for 2 years. The front cover details the intended cohort. You must make sure you use the live assignment that relates to the student's cohort for assessment and submit in the period in which the assignments are live. For example, a cohort beginning a 2-year course in September 2026 should use the set of assignments marked as being for 2026-2028 so that whatever order assignments are taken in, they will be able to re-submit improved work on the same NEA assignment if they wish to during their study of the qualification.

	Centres should avoid allowing new cohorts to use assignments which have already been live for a year, e.g. students who start the course in September 2027 using assignments for the 2026-2028 cohorts. Centres must have suitable controls in place to ensure that NEA assignment work is completed by each student independently and must not allow previously completed work for assignments which are still live to be shared as examples with other students.
Exam series each year	• January • June
Exam resits	Students can resit each examined unit twice before they complete the qualification.
NEA Submission	There are two windows each year to submit NEA outcomes and request a moderation visit by an OCR Assessor. You must make unit entries for students before you can submit outcomes to request a visit. All dates are on our administration pages.
Resubmission of students' NEA work	If students have not performed at their best in the NEA Assignments, they can improve their work and submit it to you again for assessment. They must have your agreement and you must be sure it is in the student's best interests. We use the term 'resubmission' when referring to student work that has previously been submitted to OCR for moderation. Following OCR moderation, a student can attempt to improve their work for you to assess and provide the final mark to us. There is one resubmission opportunity per NEA assignment. All work submitted (or resubmitted) must be based on the assignment that is live for assessment. For information about feedback see **Section 6**. The final piece of work must be completed solely by the student and teachers must not detail specifically what amendments should be made.
Grading	Information about unit and qualification grading is in **Section 5**.

2.3 Qualification structure

Key to units for these qualifications:

M = Mandatory	Students must complete these units.
O = Optional	Students must complete some of these units.
E = External assessment	We set and mark the exams.
N = NEA	We set the assignment. You assess the assignment and we moderate it.

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate)

For this qualification, students must complete two units:

- One mandatory externally assessed unit
- One mandatory NEA unit

Unit no	Unit title	Unit ref no (URN)	Guided learning hours (GLH)	How is it assessed?	Mandatory or optional
F160	Fundamentals of application development	M/651/0812	75	E	M
F162	Designing and communicating UX/UI solutions	T/651/0823	75	N	M

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate)

For this qualification, students must complete five units:

- Two mandatory externally assessed units
- One mandatory NEA unit
- Two optional NEA units

Unit no	Unit title	Unit ref no (URN)	Guided learning hours (GLH)	How is it assessed?	Mandatory or optional
F160	Fundamentals of application development	M/651/0812	75	E	M
F161	Developing application software	F/651/0818	70	E	M
F162	Designing and communicating UX/UI solutions	T/651/0823	75	N	M
F163	Game Development	D/651/0826	70	N	O
F164	Website Development	J/651/0829	70	N	O
F165	Immersive technology solution development	M/651/0830	70	N	O
F166	Software development	Y/651/0833	70	N	O

2.4 Purpose statement – Certificate

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate)

Qualification number: 610/3974/3

Overview

Who this qualification is for

The OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate) is for students aged 16-19 years old. It will develop knowledge, understanding and skills that will help prepare you for progression to undergraduate study when taken alongside other qualifications and are relevant to the ICT practitioners sector.

You might be interested in this qualification if you want a small qualification to take alongside and enhance your A Level studies, that builds applied or practical skills. You will have the opportunity to apply what you learn to real-life contexts, such as:

- Interpreting client requirements and documenting ideas.
- Planning and designing UX/UI solutions.
- Communicating UX/UI solutions to clients.

The qualification will also help you develop independence and confidence in using skills that are relevant to the sector and that prepare you for progressing to university courses where independent study skills are needed. You will develop the following transferable skills that can be used in both higher education and other life and work situations:

- Communicating effectively with individuals or groups. Communicating effectively with clients and other stakeholders is important in the ICT practitioners sector. It is also a vital life-skill and important for progressing to and in, higher education.
- Creativity. You will demonstrate creativity when exploring and generating ideas, making connections to find imaginative solutions and outcomes that are of value.
- Critical thinking and problem solving. You will explore the options, tools and techniques to tackle problems and use critical thinking skills to select the most appropriate way to proceed. You will plan and design solutions, checking the outcome to see if the problem has been resolved.
- Independent learning. You will spend time outside of lessons learning how to use different software packages to create solutions to problems.
- Time management. It is important both in higher education and the ICT practitioners sector that projects are delivered on time. You will learn how to use project planning tools to effectively plan projects.

This qualification will complement other learning that you're completing at Key Stage 5. If you are a full-time student, it will be part of your studies along with your A Levels.

What you will study when you take this qualification

Through a combination of theoretical study and hands-on experience, you will develop the necessary knowledge and skills that can support progression to higher education computing study.

In the examined unit, you will study key knowledge and understanding relevant to application development. In the non examined assessment (NEA) unit, you will demonstrate knowledge and skills you learn by completing a practical assignment. More information about the knowledge and skills you will develop is below.

All units in the qualification are mandatory. You must take **all** of these units:

- F160: Fundamentals of application development

  This unit is assessed by an exam.

  In this unit you will learn about the different stages that developers go through to produce a working software application, how developers scope application requirements, and the design features which make applications intuitive for users. Topics include:

  - Topic Area 1 Types of software used in application design
  - Topic Area 2 Software development models
  - Topic Area 3 Planning application development projects
  - Topic Area 4 Application design scoping
  - Topic Area 5 Human computer interface and interaction
  - Topic Area 6 Job roles and skills

- F162: Designing and communicating UX/UI solutions

  This unit is assessed by an assignment.

  In this unit you will learn the principles of UX/UI design and what makes an interface easy to use. You will learn tools and techniques to plan UX/UI solutions and how to design high-fidelity prototypes of UX/UI solutions. You will also learn how to communicate effectively with clients. Topics include:

  - Topic Area 1 Principles of UX and UI design
  - Topic Area 2 Plan UX/UI solutions
  - Topic Area 3 Design UX/UI solutions
  - Topic Area 4 Communicate UX/UI solutions
  - Topic Area 5 Review and improve UX/UI solutions

The subjects that complement this course

These subjects might complement this qualification:

- A Level Art and Design
- A Level Business Studies
- A Level Computer Science
- A Level Design and Technology
- A Level Maths
- A Level Media Studies

The types of courses you may progress to

Both the subject-specific knowledge, understanding and skills, and broader transferable skills developed in this qualification will help you progress to further study in related areas such as:

- Computer Science
- Computing
- User Experience Design.

Why you should take the OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate)

There are two qualifications available in application development. These are:

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate) – this is 150 GLH in size

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate) – this is 360 GLH in size

You should take this Certificate qualification if you want a small Level 3 qualification that builds some applied knowledge and skills in application development. This qualification is an Alternative Academic Qualification that is the same size as an AS Level qualification. It is half the size of an A Level. It could be taken alongside A Levels to help enhance your learning as it will complement A Levels, helping you to build broader knowledge and skills that are valued in undergraduate study, and relevant for progression to higher education. You would take this qualification alongside A Levels as part of your programme of study at Key Stage 5.

More information

More information about the OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate) is in these documents:

- Specification: <<insert link>>
- Sample Assessment Material (SAM) Question Papers:
  - Unit F160: <<insert link>>
- Guides to our SAM Question Papers:
  - Unit F160: <<insert link>>
- SAM Set Assignment:
  - Unit F162: <<insert link>>
- Student Guide to NEA Assignments: <<insert link>>

2.5 Purpose statement – Extended Certificate

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate)

Qualification number: 610/3975/5

Overview

Who this qualification is for

The OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate) is for students aged 16-19 years old. It will develop knowledge, understanding and skills that will help prepare you for progression to undergraduate study and are relevant to the ICT practitioners sector.

You might be interested in this qualification if you want to apply what you learn to practical, real-life contexts, such as:

- Interpreting client requirements and documenting ideas.
- Planning and designing UX/UI solutions.
- Communicating UX/UI solutions to clients.
- Planning, designing and creating two different applications.
- Testing and reviewing created applications.

The qualification will also help you develop independence and confidence in using skills that are relevant to the sector and that prepare you for progressing to university courses where independent study skills are needed. You will develop the following transferable skills that can be used in both higher education and other life and work situations:

- Communicating effectively with individuals or groups. Communicating effectively with clients and other stakeholders is important in the ICT practitioners sector. It is also a vital life-skill and important for progressing to and in, higher education.
- Creativity. You will demonstrate creativity when exploring and generating ideas, making connections to find imaginative solutions and outcomes that are of value.
- Critical thinking and problem solving. You will explore the options, tools and techniques to tackle problems and use critical thinking skills to select the most appropriate way to proceed. You will plan and design solutions, checking the outcome to see if the problem has been resolved.
- Independent learning. You will spend time outside of lessons learning how to use different software packages to create solutions to problems.
- Time management. It is important both in higher education and the ICT practitioners sector that projects are delivered on time. You will learn how to use project planning tools to effectively plan projects.

This qualification will complement other learning that you're completing at Key Stage 5. If you are a full-time student, it will be part of your studies along with A Levels.

What you will study when you take this qualification

Through a combination of theoretical study and hands-on experience, you will develop the necessary knowledge and skills that can support progression to higher education computing study.

In the examined units, you will study key knowledge and understanding relevant to application development. In the non examined assessment (NEA) units, you will demonstrate knowledge and skills you learn by completing applied or practical assignments. More information about the knowledge and skills you will develop is below.

The qualification has three mandatory units and four optional units.

These are the **mandatory** units – you must take **all** these units:

- F160: Fundamentals of application development

  This unit is assessed by an exam.

  In this unit you will learn about the about the different stages that developers go through to produce a working software application, how developers scope application requirements, and the design features which make applications intuitive for users. Topics include:

  - Topic Area 1 Types of software used in application design
  - Topic Area 2 Software development models
  - Topic Area 3 Planning application development projects
  - Topic Area 4 Application design scoping
  - Topic Area 5 Human computer interface and interaction
  - Topic Area 6 Job roles and skills

- F161: Developing application software

  This unit is assessed by an exam.

  In this unit you will learn about implementation methodology and the areas that need to be considered when applications are being developed for different platforms. You will also learn about how data moves in applications and beyond, and how to make sure applications are safe to use and the data they hold is secure. You will also learn how developers deploy finished applications to users, how they're installed on devices, and maintained in the future. Topics include:

  - Topic Area 1 Application software considerations
  - Topic Area 2 Data and flow in application software
  - Topic Area 3 API and protocols
  - Topic Area 4 Application software security
  - Topic Area 5 Operational considerations
  - Topic Area 6 Legal considerations

- F162: Designing and communicating UX /UI solutions

  This unit is assessed by an assignment.

  In this unit you will learn the principles of UX/UI design and what makes an interface easy to use. You will learn tools and techniques to plan UX/UI solutions and how to design high-fidelity prototypes of UX/UI solutions. You will also learn how to communicate effectively with clients. Topics include:

  - Topic Area 1 Principles of UX and UI design
  - Topic Area 2 Plan UX/UI solutions
  - Topic Area 3 Design UX/UI solutions
  - Topic Area 4 Communicate UX/UI solutions
  - Topic Area 5 Review and improve UX/UI solutions

These are **optional** units – you must take **two** of these units:

- F163: Game development

  This unit is assessed by an assignment.

  In this unit you will learn how types and genres of digital games and their characteristics affect game design. You will then learn how to plan, design, create, and test game prototypes. Topics include:

  - Topic Area 1 Game design
  - Topic Area 2 Plan and design high-fidelity game prototypes
  - Topic Area 3 Create high-fidelity game prototypes
  - Topic Area 4 Test high-fidelity game prototypes
  - Topic Area 5 Review and improve high-fidelity game prototypes

- F164: Website development

  This unit is assessed by an assignment.

  In this unit you will learn about website principles and the components of web pages. You will then learn how to plan, design, create, and test website prototypes that can be viewed on a range of devices. Topics include:

  - Topic Area 1 Fundamentals of website development
  - Topic Area 2 Plan and design high-fidelity website prototypes
  - Topic Area 3 Create high-fidelity website prototypes
  - Topic Area 4 Test high-fidelity website prototypes
  - Topic Area 5 Review and improve the effectiveness of high-fidelity website prototypes

- F165: Immersive technology solution development

  This unit is assessed by an assignment.

  In this unit you will learn the principles of immersive technologies. You will then learn how to plan, design, create, and test immersive technology solution prototypes. Topics include:

  - Topic Area 1 Principles of immersive technology
  - Topic Area 2 Plan and design high-fidelity immersive technology solution prototypes
  - Topic Area 3 Create high-fidelity immersive technology solution prototypes
  - Topic Area 4 Test high-fidelity immersive technology prototypes
  - Topic Area 5 Review and improve the effectiveness of high-fidelity immersive technology prototypes

- F166: Software development

  This unit is assessed by an assignment.

  In this unit you will learn about software design principles and different programming language types. You will then learn how to design, create, and test software solutions. Topics include:

  - Topic Area 1 Fundamentals of software development
  - Topic Area 2 Design software solutions
  - Topic Area 3 Create software solutions
  - Topic Area 4 Test software solutions
  - Topic Area 5 Review and improve software solutions

The subjects that complement this course

These subjects might complement this qualification:

- A Level Art and Design
- A Level Business Studies
- A Level Computer Science
- A Level Design and Technology
- A Level Maths
- A Level Media Studies

The types of courses you may progress to

Both the subject-specific knowledge, understanding and skills, and broader transferable skills developed through these units, will help you progress to further study in related areas such as:

- Computer Science
- Computer Games Development
- Computing
- Creative Computing
- Web and Mobile Development
- Web and User Experience Design

Why you should take the OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate)

There are two qualifications available in application development. These are:

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate)– this is 150 GLH in size

OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate) – this is 360 GLH in size

You should take this Extended Certificate qualification if you want a Level 3 qualification that builds applied knowledge and skills in application development. This qualification is an Alternative Academic Qualification that is the same size as an A Level. When it is taken alongside A Levels it will complement them, helping you to build broader knowledge and skills that are valued in undergraduate study, and relevant for progression to higher education. You would take this qualification alongside A Levels as part of your programme of study at Key Stage 5.

More information

More information about the Cambridge Advanced National in Computing: Application Development (Extended Certificate) is in these documents:

- Specification: <<insert link>>
- Sample Assessment Material (SAM) Question Papers:
  - Unit F160: <<insert link>>
  - Unit F161: <<insert link>>
- Guides to our SAM Question Papers:
  - Unit F160: <<insert link>>
  - Unit F161: <<insert link>>
- SAM Set Assignment(s):
  - Unit F162: <<insert link>>
  - Unit F163: <<insert link>>
  - Unit F164: <<insert link>>
  - Unit F165: <<insert link>>
  - Unit F166: <<insert link>>
- Student Guide to NEA Assignments: <<insert link>>

3 About these qualifications

3.1 Qualification size

The size of each qualification is described in terms of Guided Learning Hours (GLH) and Total Qualification Time (TQT).

GLH indicates the approximate time (in hours) you will spend supervising or directing study and assessment activities. We have worked with people who are experienced in delivering related qualifications to determine the content that needs to be taught and how long it will take to deliver.

TQT includes two parts:

- GLH
- an estimate of the number of hours a student will spend on unsupervised learning or assessment activities (including homework) to successfully complete their qualification.

The OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate) is 150 GLH and 200 TQT.

The OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate) is 360 GLH and 500 TQT.

3.2 Availability and language

The Level 3 Alternative Academic Qualification Cambridge Advanced Nationals are available in England only. They are **not** available in Wales or Northern Ireland.

The qualifications and their assessment materials are available in English only. We will only assess answers written in English.

3.3 Prior knowledge and experience

Recognition of prior learning (RPL) is the process for recognising learning that never received formal recognition through a qualification or certification. It includes knowledge and skills gained in school, college or outside of formal learning situations. These may include:

- domestic/family life
- education
- training
- work activities
- voluntary activities.

In most cases RPL will not be appropriate for directly evidencing the requirements of the NEA assignments for the Cambridge Advanced National qualifications. However, if you feel that your student could use RPL to support their evidence, you must follow the guidance provided in our **RPL Policy**.

4 Units

4.1 Guidance on unit content

This section describes what must be taught so that students can access all available marks and meet assessment criteria.

4.1.1 Externally assessed units (F160 and F161)

The externally assessed units contain a number of topic areas.

For each topic area, we list the **teaching content** that must be taught and give information on the **breadth and depth** of teaching needed.

Teaching content

Questions can be asked about anything in the teaching content or breadth and depth columns.

Breadth and depth

The breadth and depth column:

- clarifies the breadth and depth of teaching needed
- indicates the range of knowledge and understanding that can be assessed in the exam
- confirms any aspects that you do not need to teach as 'does not include' statements.

Teaching must cover **both** the **teaching content** and **breadth and depth** columns.

Knowledge and understanding

This is what we mean by knowledge and understanding:

Knowledge	• Be able to identify or recognise an item, for example on a diagram. • Use direct recall to answer a question, for example the definition of a term.
Understanding	• To assess and evidence the perceived meaning of something in greater depth than straight identification or recall. • Understanding will be expressed and presented using terms such as: how; why; when; reasons for; advantages and disadvantages of; benefits and limitations of; purpose of; suitability of; recommendations for improvement; appropriateness of something to/in different contexts.

Students will need to **understand** the content, unless the breadth and depth column identifies it as knowledge only.

Any item(s) that should be taught as **knowledge** only will start with the word 'know' in the breadth and depth column.

All other content must be taught as understanding.

4.1.2 NEA units (F162 - F166)

The NEA units contain a number of topic areas.

For each topic area, we list **teaching content** that must be taught and give **exemplification**. The exemplification shows the teaching expected to equip students to successfully complete their assignments.

4.1.3 Command words

Appendix B gives information about the command words that will be used in the external assessments and the NEA assessment criteria.

4.1.4 Performance objectives (POs):

Each Cambridge Advanced National qualification has four Performance Objectives.

PO1	Show knowledge and understanding
PO2	Apply knowledge and understanding
PO3	Analyse and evaluate knowledge, understanding and performance
PO4	Demonstrate and apply skills and processes relevant to the subject

PO1 is assessed in the externally assessed unit only.

PO4 is assessed in the NEA units only.

The weightings of the Performance Objectives across the units in the **Certificate** qualification are:

Performance Objective	**Externally Assessed unit (range)**	**NEA units**	**Overall weighting**
PO1	12.5 – 20.8%	n/a	12.5 - 20.8%
PO2	16.7 - 25%	14.6%	31.3 - 39.6%
PO3	12.5%	10.4%	22.9%
PO4	n/a	25%	25%
Overall weighting of assessments	**50%**	**50%**	**100%**

The weightings of the Performance Objectives across the units in the **Extended Certificate** qualification are:

Performance Objective	**Externally Assessed unit (range)**	**NEA units**	**Overall weighting**
PO1	10 - 16.7%	n/a	10 - 16.7%
PO2	13.3 - 20%	15 - 15.8%	28.3 - 35.8%
PO3	10%	13.3 - 14.2%	28.3 - 24.2%
PO4	n/a	30 - 31.7%	30 - 31.7%
Overall weighting of assessments	**40%**	**60%**	**100%**

4.2 Externally assessed units

4.2.1 Unit F160: Fundamentals of application development

Unit aim

Software applications are all around us, and each one has been carefully designed to perform a specific function for the end user. This unit develops your knowledge and understanding of the stages of software application development and how these applications are designed.

In this unit you will learn what an application is and the tasks/functions they're commonly developed to do. You will learn about the different stages that developers go through to produce a working software application, how developers scope application requirements, and the design features which make applications intuitive for users. Finally, you will learn about the job roles available in application development and some of the skills required for these roles.

Unit F160: Fundamentals of application development	
Topic Area 1: Types of software used in application design	
Teaching content	**Breadth and depth**
1.1 Programs and applications	
□ Programs □ Applications	**To include:** □ Know what a program is □ Know what an application is □ Know the characteristics of a program □ Know the characteristics of an application □ Know the function of an application □ The relationship between programs and applications □ Know the different types of device that use programs/applications
1.2 Operating Systems (OS) for application software	
□ Network □ Open OS □ Proprietary	To include: □ Know the characteristics of each type of operating system used to run application software □ The advantages and disadvantages of each type of operating system □ Know the types of device that use each type of operating system □ How defined client requirements affect the selection of an operating system
1.3 Application types and categories	
1.3.1 Application types □ Communication □ Educational □ Entertainment □ Games □ Lifestyle □ Productivity □ Protection and utility □ Web browsers	To include: □ The purpose of each application type □ The characteristics of each application type
1.3.2 Application software categories □ Open □ Closed □ Shareware □ Freeware □ Embedded	To include: □ Know the characteristics of each application software category □ The purpose of each application software category

	□ Know the types of device that use each application software category □ The advantages and disadvantages of each application software category □ How defined client requirements affect the selection of an appropriate application software category
1.3.3 Application software types □ Off-the-Shelf □ Custom Off-the-Shelf □ Bespoke	To include: □ Know characteristics of each application software type □ The purpose of each application software type □ The advantages and disadvantages of each software application type □ How defined client requirements affect the selection of an appropriate application software type
Topic Area 2: Software development models	
Teaching content	**Breadth and depth**
2.1 Software development models	
□ Traditional model • Waterfall □ Prototyping model • Rapid Throwaway • Incremental • Evolutionary □ Iterative model • Rapid Application Development (RAD) • Spiral • Agile	To include: □ Know the characteristics of each software development model □ Know why software development models are used □ The advantages and disadvantages of using software development models □ The diagrammatical representation of each software development model □ The advantages and disadvantages of each software development model □ The type of development for which each software development model is used □ How the type of development determines the software development model used
2.2 The common phases of software development models	
□ Planning • Requirements • Feasibility □ Design □ Constructing/creation □ Testing □ Implementation • Phased • Parallel • Big bang (crash) □ Documentation creation □ Maintenance	To include: □ Know the common phases included in the software development models □ Know the tasks included in each phase in software development models □ How the phases interact and iterate in software development models □ The importance of interaction and iteration between the phases
Topic Area 3: Planning application development projects	
Teaching content	**Breadth and depth**
3.1 Planning projects	
□ Purpose of planning projects □ Planning considerations • Budget • Constraints • Legislation ○ Copyright	To include: □ Why planning application development projects is important □ The advantages and disadvantages of planning application development projects

○ Data protection ○ Electronic communications • Resources • Success criteria • Time	□ The consequences of not planning application development projects □ The importance of each planning consideration □ How each planning consideration impacts application development Does not include: □ Knowledge of details of specific acts
3.2 Project planning tools	
□ Arrow diagram □ Critical Path Analysis (CPA)/Critical Path Method (CPM) □ Flowchart □ Gantt charts □ PERT charts □ Strengths/Weaknesses/Opportunities/Threats (SWOT) analysis	To include: □ Know the components and conventions of each project planning tool □ The advantages and disadvantages of each project planning tool □ How defined client requirements determine/affect the selection of project planning tools □ Use of project planning tools for effective planning
Topic Area 4: Application design scoping	
Teaching content	**Breadth and depth**
4.1 Methods of gathering client requirements	
□ Document analysis □ Focus group □ Interviews □ Meetings □ Observation □ Problem reports □ Questionnaire □ Shadowing □ Suggestion analysis	To include: □ Know the purpose of each method □ When each method is used □ Know the type of information and data that can be collected using each method □ The advantages and disadvantages of using each method □ How defined client requirements determine the method used
4.2 Client requirement specifications	
□ Purpose of new system □ Functional requirements □ Non-functional requirements □ Process constraints □ Current system deficiencies □ Data formats □ Client defined constraints • Budget • Time • Integration • Software • Hardware • Data storage location ○ Local/onsite ○ Cloud ○ Physical storage devices □ Version and source control	To include: □ The importance of creating client requirement specifications □ Know the elements of client requirement specifications □ The purpose of each element □ How each requirement could be gathered □ How to elicit client requirements
4.3 Decomposition methods	
□ Abstraction □ Pattern recognition □ Modularisation • Top down • Bottom up □ Parsing of requirements	To include: □ Know the purpose of each decomposition method □ When it is appropriate to use each decomposition method

	□ The advantages and disadvantages of each decomposition method □ How gathered client requirements affect the selection of decomposition methods □ Use of decomposition methods to visualise application designs
Topic Area 5: Human computer interface and interaction	
Teaching content	**Breadth and depth**
5.1 Human computer interaction and devices	
5.1.1 Types of human computer interaction □ Audio □ Movement/gesture □ Touch □ Visual • Command line • GUI	To include: □ The purpose of each type of user interaction □ The different types of user interactions □ Know the type of device on which each type of interaction is used □ The advantages and disadvantages of each type of interaction used with application software □ How gathered client requirements affect the selection of interaction types
5.1.2 Types of device □ Desktop □ Games console □ Laptop □ Smart speaker □ Smart TV □ Smartphone □ Tablet □ Augmented Reality (AR)/Virtual Reality (VR)/Mixed Reality (MR) devices	To include: □ Know each type of device that uses application software □ The characteristics of each type of device
5.2 Human computer interface visual design considerations	
□ Colours □ Interaction □ Location hierarchy □ Messages • Help • Error □ Typography • Style • Size	To include: □ How each visual design consideration is used in the design of human computer interfaces □ How to improve the effectiveness of human computer interfaces □ How gathered client requirements impact visual design
5.3 Human computer interface design documents and diagrams	
□ Processing and data handling • Data flow diagrams o Level 0 o Level 1 • Flowcharts □ User interface designs • Visualisation diagram • Wireframe diagrams	To include: □ Know the components and conventions of each document and diagram □ When each document and diagram is appropriate for use □ How to create each document and diagram □ What makes each document and diagram effective □ How to improve the effectiveness of documents and diagrams for users

Topic Area 6: Job roles and skills	
Teaching content	**Breadth and depth**
6.1 Job roles	
□ Application Designer □ Mobile Application Designer □ Project Manager □ Systems Analyst □ Systems Designer □ User Experience Designer (UXD) □ User Interface Designer (UID)	To include: □ Know the main responsibilities of each job role related to software application development □ How each job role contributes to software application development
6.2 Communication skills required in application development	
□ Appropriate language to meet the needs of the audience □ Non-verbal □ Questioning techniques to elicit specific information □ Verbal □ Written	To include: □ Know the characteristics of each communication skill □ How each communication skill contributes to software application development □ Appropriate use of each communication skill • Job role • Stage in application development

Assessment guidance

This unit is assessed by an exam. The exam is 1 hours and 15 minutes and has **60** marks in total. All questions in the exam are compulsory.

The exam will **always** have:

A short scenario	• This will develop through the paper.
Questions to assess Performance Objectives 1, 2, and 3	• PO1: these questions will require students to recall generic knowledge and understanding. • PO2: these questions will require students to apply knowledge and understanding. • PO3: these questions will require students to analyse and evaluate knowledge, understanding and performance in relation to the scenario.
A range of question types	• Forced choice/controlled response questions. • Short answer, closed response questions. • Extended constructed response questions with points-based marks schemes. • Extended constructed response questions with levels of response marks schemes. • One six mark and one nine mark extended constructed response question with a levels of response marks scheme.
Questions relating to each Topic Area	• Content will be sampled from all topic areas, with at least one question or part question relating to each topic area.

This will be conducted under examination conditions. For more details refer to the **Administration area.**

The Application Development **Guide to our Sample Assessment Material** gives more information about the layout and expectations of the exam.

The exam for this unit assesses the following Performance Objectives:

- PO1 – Show knowledge and understanding
- PO2 – Apply knowledge and understanding
- PO3 – Analyse and evaluate knowledge, understanding and performance.

Synoptic assessment

This unit allows students to gain underpinning knowledge and understanding relevant to the qualification and sector. The NEA units draw on and strengthen this learning with students applying their learning in a practical way.

The following NEA units have synoptic links with this unit. The synoptic grids at the end of these NEA units show these synoptic links.

- F162: Designing and communicating UX /UI solutions
- F163: Game development
- F164: Website development
- F165: Immersive technology solution development
- F166: Software development

More information about synoptic assessment in these qualifications can be found in **Section 5.2 Synoptic Assessment.**

4.2.2 Unit F161: Developing application software

Unit aim

Applications come in many different forms with some operating on a single platform and others functioning across many platforms. This unit develops your knowledge and understanding of the considerations needed to implement, commission, and maintain secure applications.

In this unit you will learn implementation methodology and the areas that need to be considered when applications are being developed for different platforms. You will also learn about how data moves in applications and beyond, and how to make sure applications are safe to use and the data they hold is secure. Finally, you will learn how developers deploy finished applications to users, how they're installed on devices, and maintained in the future.

Unit F161: Developing application software	
Topic Area 1: Application software considerations	
Teaching content	**Breadth and depth**
1.1 Application platforms	
□ Augmented Reality (AR)/Virtual Reality (VR)/Mixed Reality (MR) • Educational • Instructional • Research □ Websites • Ecommerce • Informative • Educational • Social media □ Computer games	To include: □ Uses of each application platform □ The advantages and disadvantages of each application platform
1.2 Devices	
□ Console □ Desktop □ Haptic □ Laptop □ Server □ Smart devices □ Tablet/hybrid □ Wearables	To include: □ The characteristics of each type of device that application platforms run on □ The advantages and disadvantages of each device
1.3 Storage locations	
1.3.1 On-Site □ File servers □ Network Attached Storage (NAS) devices □ Portable storage devices □ Solid State Drive (SSD) □ Storage Area Network (SAN)	To include: □ The characteristics of each storage location □ The advantages and disadvantages of each storage location □ The factors to consider when selecting storage locations
1.3.2 Cloud storage □ Location of cloud storage • Private • Public • Hybrid • Community □ Types of cloud storage • File storage • Object storage • Block storage • Elastic/scalable storage • Cloud-based database services	To include: □ The characteristics of each cloud storage location □ The advantages and disadvantages of each cloud storage location □ The factors to consider when selecting cloud storage locations □ The characteristics of each cloud storage type □ The advantages and disadvantages of each cloud storage type □ The factors to consider when selecting cloud storage types

Topic Area 2: Data and flow in application software	
Teaching content	**Breadth and depth**
2.1 Data format and types	
□ Data formats • American Standard Code for Information Interchange (ASCII) • Unicode • Comma-separated Values (CSV) • Fixed width • JavaScript Object Notation (JSON) • Extensible Markup Language (XML) □ Data types • Boolean • Character • Date • Integer • Real • String	To include: □ Know the characteristics of each data format □ How each data format is used □ The advantages and disadvantages of each data format □ Know the characteristics of each data type □ How each data type is used □ The advantages and disadvantages of each data type
2.2 Data flow	
□ Input • Number • Text • Movement • Audio • Image ○ Moving ○ Static □ Storage • On-site • Cloud □ Output information • Number • Text • Movement • Audio • Image ○ Moving ○ Static □ Black box concept • Flow in • Flow to storage • Flow out	To include: □ Know the difference between data and information □ How data is converted to information □ How data flows through application software □ The types of data that flow through application software □ How information flows from application software □ The types of information that flow from application software □ The storage locations required for application software □ How to diagrammatically represent data flow using black box concept Does not include: □ The processing/programming required to convert data to information
2.3 Data States	
□ At rest □ In transit (motion) □ In use	To include: □ The characteristics of each data state □ When each state is used
Topic Area 3: API and protocols	
Teaching content	**Breadth and depth**
3.1 Application Programming Interfaces (API)	
□ Role □ Types • Composite • Internal • Private • Public • Partner □ Architecture	To include: □ The role of APIs and their use □ When each API type is used □ The advantages and disadvantages of each API type □ When each API architecture is used □ The advantages and disadvantages of each API architecture

• Representational State Transfer (REST) • Simple Object Access Protocol (SOAP) • Remote Procedure Call (RPC)	
3.2 Protocols	
□ File Transfer Protocol (FTP) □ Hyper Text Transfer Protocol (HTTP) □ Post Office Protocol (POP) □ Simple Mail Transport Protocol (SMTP) □ Simple Network Management Protocol (SNMP) □ Transport Control Protocol (TCP) □ User Datagram Protocol (UDP) □ Internet Control Message Protocol (ICMP) □ Internet Protocol (IP)	To include: □ The structure, content, and use of the 4-layer TCP/IP stack □ Know the role of each protocol □ When each protocol is used Does not include: □ OSI model
Topic Area 4: Application software security	
Teaching content	**Breadth and depth**
4.1 Security considerations	
□ Threats • Botnets • Denial of Service (DOS)/Distributed Denial of Service (DDoS) • Hacking • Lack of supplier support • Malicious spam • Malware • Out of date ○ Software ○ Hardware ○ Firmware □ Physical security mitigations • Biometrics • Cable locks • Cameras • Locks • RFID • Safe • Swipe cards □ Digital security mitigations • Access rights • Anti-malware • Back-up • Cryptography • Encryption ○ At rest ○ In transit • Firewalls ○ Hardware ○ Software • Two-Factor Authentication (2FA)	To include: □ Know current threats to application security □ The risk(s) to application security posed by each current threat □ Know current physical and digital security mitigations □ How current physical and digital security mitigations protect application software from threats Does not include: □ The details of specific threats □ The details of the specific workings of mitigations
Topic Area 5: Operational considerations	
Teaching content	**Breadth and depth**
5.1 Testing	
□ Test plan structure • Test number • Test type • Test description ○ Purpose	To include: □ The purpose of testing □ The importance of testing □ The impact of not testing on applications

○ Procedure • Test data • Expected result • Actual result • Remedial action required • Retest result □ Types of test data • Normal • Extreme • Erroneous □ Types of testing • Technical • User	□ The advantages and disadvantages of testing □ The structure and contents of test plans □ The importance of testing during the development of applications □ The importance of remedial action and retesting □ Know what each type of test data is □ The role of each type of test data during testing □ Know the purpose of each type of testing □ The advantages and disadvantages of each type of testing □ When each type of testing should take place □ How each type of testing takes place Examples of **technical testing** may include: □ Fuzz testing □ Load/stress testing □ Migration testing
5.2 Types of application software Installation	
□ Create ghost/image and deployment □ Upgrade □ Clean install □ Repair/modify installs □ Remote install □ Unattended installation □ Cloud download/install □ Mobile install □ Network install	To include: □ How the different installation processes are completed □ The advantages and disadvantages of the different installation processes □ When it is appropriate to use each installation process Does not include: □ Completing software installations
5.3 Policies	
□ Application user guide □ Acceptable Use Policy (AUP) □ Backup(s) □ Codes of practice □ Staying safe online □ Use of information	To include: □ Know the purpose and content of each policy to be considered when developing application platforms □ How each policy is applied when developing application platforms Does not include: □ Creating new policies
Topic Area 6: Legal considerations	
Teaching content	**Breadth and depth**
6.1 Legal considerations	
Legislations and regulations □ Computer Misuse Act (CMA) □ Data Protection Act (DPA) □ UK General Data Protection Regulation (UK GDPR) □ Freedom of Information Act (FOIA) □ Privacy and Electronic Communications Regulations (PECR) Independent bodies □ Information Commissioner's Office (ICO) in the UK	To include: □ Know the latest version of each act/regulation □ Know the main purpose(s) of each act/regulation □ The actions that must be taken to comply with each act/regulation when developing application software □ The impact of non-compliance with each act/regulation □ How PECR relate to DPA and UK GDPR

	□ The role of Information Commissioner's Office (ICO) in the UK Does not include: □ Knowing the detailed content of each act/regulation

Assessment guidance

This unit is assessed by an exam. The exam is 1 hours and 15 minutes and has **60** marks in total. All questions in the exam are compulsory.

The exam will **always** have:

A short scenario	• This will develop through the paper.
Questions to assess Performance Objectives 1, 2, and 3	• PO1: these questions will require students to recall generic knowledge and understanding. • PO2: these questions will require students to apply knowledge and understanding. • PO3: these questions will require students to analyse and evaluate knowledge, understanding and performance in relation to the scenario.
A range of question types	• Forced choice/controlled response questions. • Short answer, closed response questions. • Extended constructed response questions with points-based marks schemes. • Extended constructed response questions with levels of response marks schemes. • One six mark and one nine mark extended constructed response question with a levels of response marks scheme.
Questions relating to each Topic Area	• Content will be sampled from all topic areas, with at least one question or part question relating to each topic area.

This will be conducted under examination conditions. For more details refer to the **Administration area.**

The **Application Development Guide to our Sample Assessment Material** gives more information about the layout and expectations of the exam.

The exam for this unit assesses the following Performance Objectives:

- PO1 – Show knowledge and understanding
- PO2 – Apply knowledge and understanding
- PO3 – Analyse and evaluate knowledge, understanding and performance.

Synoptic assessment

This unit allows students to gain underpinning knowledge and understanding relevant to the qualification and sector. The NEA units draw on and strengthen this learning as students will apply their learning to practical tasks.

The following NEA units have synoptic links with this unit. The synoptic grids at the end of these NEA units show these synoptic links.

- F162: Designing and communicating UX /UI solutions
- F163: Game development
- F164: Website development
- F165: Immersive technology solution development
- F166: Software development.

More information about synoptic assessment in these qualifications can be found in **Section 5.2 Synoptic Assessment.**

4.3 NEA Units

4.3.1 Unit F162: Designing and communicating UX/UI solutions

Unit Aim

Applications are all around us and behind every successful application, a process has been completed to enhance the way it looks and to optimise interactivity. Every screen, button, and other visual components you see when using an application should form a carefully crafted user interface that provides an intuitive and straight forward experience for users. Principles of user experience (UX) and user interface (UI) design are vital in application development and if they're not followed properly, applications could be unusable.

In this unit you will learn the principles of UX/UI design: how users interact with applications, how applications should adapt to support specific user needs, and what makes interfaces interesting and easy to use. You will learn how to develop ideas for UX/UI solutions which meet specific requirements and the design process to create graphical representations of these. Communication skills are vital in the IT sector and in this unit, you will also learn how to prepare a "showcase" to demonstrate UX/UI solutions to clients.

Unit F162: Designing and communicating UX/UI solutions	
Topic Area 1: Principles of UX and UI design	
Teaching content	**Exemplification**
1.1 Basics of UX and UI	
□ User experience (UX) design □ User Interface (UI) design	To include: □ The role, and importance, of UX and UI design in application development □ How UX and UI design interrelate
1.2 Application end user considerations	
□ Experience • Novice/beginner • Occasional • Regular • Expert user □ Available hardware • Input devices • Screen sizes • Type of device □ Accessibility needs	To include: □ How each consideration impacts UX/UI design Examples of **accessibility needs** may include: □ Visual impairments □ Motor difficulties □ Cognitive impairments or learning disabilities □ Hearing impairments
1.3 UX/UI design principles	
□ Perception □ Navigation design principles • Hierarchy • Menu selection • Recognition vs recall □ Schneiderman's 8 Golden Rules of interface design • Consistency • Enable shortcuts • Include informative feedback • Dialogue yields closure • Simple error handling • Easy reversal of actions • Support internal locus of control • Reduce short-term memory load □ Interface layout design principles • Above and below the fold	To include: □ Why user perception is important within UX/UI design □ How user perception impacts UX/UI design □ How each navigation design principle impacts UX/UI design □ How Schneiderman's 8 Golden Rules of interface design impacts UX/UI design □ How interface layout design principles impact UX/UI design

• Colour theory • Information visualisation • Principle of thirds • Typography	
1.4 UX/UI design psychology	
□ Cognitive load □ Hicks law □ Law of Proximity	To include: □ The key features and characteristics of each principle □ Why each principle is important to UX/UI design □ How each principle impacts UX/UI design Does not include: □ Mathematical modelling in Hicks law Other examples of **UX/UI design psychology** may include: □ Von Restorff effect □ Serial position effect
1.5 UX/UI experience	
1.5.1 Factors that impact UX □ Accessible □ Creditable □ Desirable □ Findable □ Usable □ Useful □ Valuable	To include: □ How each factor impacts UX design
1.5.2 Features of UI □ Types of UI • Command line interface (CLI) • Form-based user interface • Graphical user interface (GUI) • Menu-driven user interface • Natural language user interface • Touch user interface • Voice user interface (VUI) □ Interaction types • Function keys • Gestures • Voice • WIMP (Windows Icons Menus Pointers)	To include: □ The features, characteristics and use of each type of UI □ The advantages and disadvantages of each type of UI □ The features, characteristics and use of each type of interaction □ The advantages and disadvantages of each type of interaction □ How users interact with each type of UI
1.6 UX/UI interface design standardisation	
□ Interface standards • Common user interface layouts, icons and labels throughout the application • Cross-platform standards • Standard interface widgets • Standard protocols	To include: □ The purpose of interface design standardisation □ The features and characteristics of each interface standard □ How interface standards impact UX/UI design

Topic Area 2: Plan UX/UI solutions	
Teaching content	**Exemplification**
2.1 Requirements of UX/UI solutions	
□ Types of requirements • Client requirements • User requirements • Solution requirements ○ Functional requirements ○ Interface requirements ○ Non-functional requirements □ Sources of UX/UI solution requirements • Client briefs • Current systems • Existing documents • Users/user profiles □ Tools to document UX/UI solution requirements • Requirements specification ○ Purpose/scope ○ Business/client requirements ○ User requirements ○ Functional requirements ○ Interface requirements ○ Non-functional requirements • Use case diagrams ○ Actors ○ System interactions	To include: □ How each type of requirement impacts the planning and design of UX/UI solutions □ How to source and identify UX/UI solution requirements □ How to decompose UX/UI solution requirements into logical components □ How to identify the required inputs and outputs when planning UX/UI solutions □ The components and conventions of tools to document UX/UI solution requirements □ How to use tools to document UX/UI solution requirements
2.2 Tools and techniques to document UX/UI ideas and design concepts	
□ Tools and techniques to document ideas • Mind map • Mood boards • Spider diagrams □ Tools and techniques to document design concepts • Low-fidelity prototypes ○ Wireframes • Paper prototyping • Sketches and diagrams	To include: □ How ideas develop into design concepts □ The purpose of each tool and technique when documenting UX/UI ideas and design concepts □ The components and conventions of each tool and technique when documenting UX/UI ideas and design concepts □ How to use tools and techniques to document ideas and design concepts for UX/UI solutions
Topic Area 3: Design UX/UI solutions	
Teaching content	**Exemplification**
3.1 Tools to represent UX/UI solutions	
3.1.1 Design tools □ Diagrams • Types ○ Flow chart ○ Navigation ○ Task flows ○ Wireflow • UX/UI design features ○ Interaction flows ○ Navigation routes ○ Steps within processes ○ User steps to complete actions □ High-fidelity prototypes • Types ○ Graphical mock-ups	To include: □ The conventions and layouts of diagrams and hi-fidelity prototypes □ How diagrams are used to show UX/UI design features □ How hi-fidelity prototypes are used to show UX/UI design features □ How to use diagrams and hi-fidelity prototypes to design UX/UI solutions

○ Screen flows ○ Interactive • UX/UI design features ○ Navigation aides ○ House style ○ Layout ○ Content ○ System interaction and event handling ○ Error handling and feedback	
3.1.2 Software tools □ Software types • Standard software • Vector drawing • Diagramming • Interface prototyping software □ Software tools and techniques • Image/canvas size • Layout tools • Drawing tools • Layers and grouping • Typography • Image library objects • Interactivity	To include: □ How to use software tools and techniques to create diagrams and high-fidelity prototypes to show UX/UI solutions
3.2 Tools and techniques to check UX/UI solution designs	
□ Method of checking • Checklist □ UI audit metrics to check • Branding and messaging • Customer journey bottlenecks and roadblocks • Design inconsistencies • Layout and hierarchy inconsistencies • Legal compliance • Usability and accessibility • Usability heuristics □ Interface metrics to check • Ability to configure the interface • Ability to navigate within the system • Keystroke effort per task	To include: □ The structure, content and use of checklists □ How to use metrics to check UX/UI solution designs
Topic Area 4: Communicate UX/UI solutions	
Teaching content	**Exemplification**
4.1 Develop UX/UI solution showcases	
□ Showcase formats □ Showcase content considerations • Type • Depth • Relevance □ Showcase design considerations • Colour scheme • Language and vocabulary • Layout • Style	To include: □ The purpose of UX/UI solution showcases □ The different formats UX/UI solution showcases take and when each is appropriate □ How to develop UX/UI solution showcases □ How UX/UI solution showcase content considerations are adapted for the intended audience □ How UX/UI solution showcase design considerations are adapted for the intended audience Examples of **showcase formats** may include: □ Live presentation (in person or remote) □ Slideshow with audio overlay

	□ Video
4.2 Techniques to deliver UX/UI solution showcases	
□ Resources required • Hardware • Software □ Techniques for effective communication • Clarity • Coherence • Completeness • Conciseness • Correctness • Courteousness	To include: □ The resources used to deliver UX/UI solution showcases □ How to use resources to deliver UX/UI solution showcases to clients □ How to use techniques for effective communication to deliver UX/UI solution showcases to clients
Topic Area 5: Review and improve UX/UI solutions	
Teaching content	**Exemplification**
5.1 Review the fitness for purpose of UX/UI solutions	
□ Suitability for meeting: • Client requirements • User requirements • Solution requirements □ Application of UX/UI design principles	To include: □ How to assess strengths and weaknesses of UX/UI solutions □ How to compare UX/UI solutions against requirements □ How to assess the application of UX/UI design principles
5.2 Improvements to UX/UI solutions	
□ User experience □ Use of UX/UI design principles □ Use of principles of UX/UI design psychology □ Use of UX/UI interface standards	To include: □ How to assess potential improvements to UX/UI solutions Does not include: □ Implementing improvements to UX/UI solutions
5.3 Review the processes used to plan, design and communicate UX/UI solutions	
□ Effectiveness of processes used □ Effectiveness of tools and techniques used	To include: □ How to assess the strengths and weaknesses in the processes used to plan, design and communicate UX/UI solutions □ How to assess the effectiveness of the processes used to plan, design and communicate UX/UI solutions □ How to assess the effectiveness of tools and techniques used to plan, design and communicate UX/UI solutions

Assessment criteria

Section 6.4 provides full information on how to assess the NEA units and apply the assessment criteria.

These are the assessment criteria for the tasks for this unit. The assessment criteria indicate what is required in each task. Students' work must show that all aspects of a criterion have been met in sufficient detail for it to be **successfully achieved** (see **Section 6.4.1**). If a student's work does not fully meet a criterion, you must not award that criterion.

The command words used in the assessment criteria are defined in **Appendix B**.

Pass	Merit	Distinction
P1: Describe the client and user requirements for the UX/UI solution.	**M1: Explain** how the functional and non-functional requirements impact the design of the UX/UI solution.	**D1: Create** use case diagrams to show users and their interactions with the UX/UI solution.
P2: Describe the functional and interface requirements for the UX/UI solution.		
P3: Document appropriate UX/UI solution ideas.	**M2: Document** an appropriate UX/UI design concept.	
P4: Create diagrams that show the interaction flows and navigation routes for the UX/UI solution.	**M3: Explain** how navigation design principles have been applied to the UX/UI solution.	
P5: Create diagrams that show the steps within processes for the UX/UI solution.	**M4: Explain** how Schneiderman's 8 Golden Rules of interface design have informed the UX/UI solution.	
P6: Create diagrams that show user steps to complete actions for the UX/UI solution.		
P7: Create a high-fidelity prototype for the UX/UI solution.		**D2: Implement** error handling and feedback appropriate for the UX/UI solution.
P8: Describe how the UX/UI solution is appropriate for users.		**D3: Assess** the UX/UI solution in relation to UX/UI design psychology.
P9: Check the UX/UI solution against audit and interface metrics.	**M5: Justify** the appropriateness of the checks.	**D4: Assess** the UX/UI solution in relation to UX/UI interface standards.
P10: Create a UX/UI showcase appropriately designed for the client.	**M6: Use** techniques for effective communication to deliver the UX/UI showcase.	
P11: Deliver a UX/UI showcase that communicates the UX/UI solution using content appropriate for the client.		
P12: Describe the strengths and weaknesses of the UX/UI solution.	**M7: Discuss** potential improvements to the UX/UI solution.	
		D5: Evaluate the effectiveness of the processes used to plan and design the UX/UI solution.

Assessment guidance

This assessment guidance gives you information relating to the assessment criteria. There might not be additional assessment guidance for each assessment criterion. It is included only where it is needed.

Assessment Criteria	Assessment guidance
P1	• Students **must** describe both the client and user requirements. Students must expand the descriptions into specific requirements which can be used as criteria to review against in Task 4. This assessment criterion could be evidenced in a requirements specification.
P2	• Students **must** describe what the UX/UI solution should be capable of doing to meet the user requirements. Students **must** also describe the requirements of the user interface. This assessment criterion **could** be evidenced in a requirements specification.
P3	• Students **must** document ideas for the UX/UI solution based on the requirements described in P1 and P2. Students **must** use at least **one** of the tools and techniques in Topic Area 2.2. Students **must** document at least **two** ideas on a mind map, for example.
M1	• Students **must** explain how the requirements detailed in P1 and P2 impact the design of the UX/UI solution. Where students do not achieve P1 and/or P2, it is still possible to achieve M1. This assessment criterion **could** be evidenced in a requirements specification.
M2	• Students **must** document the UX/UI design concept using at least **one** of the tools and techniques in Topic Area 2.2. There **must** be a clear relationship between the ideas documented in P3 and the UX/UI design concept. Where students do not achieve P3 it is still possible to achieve M2. The UX/UI design concept **must** be appropriate for the client and user requirements.
D1	• Students **must** create use case diagrams which show how users specified in the scenario will interact with the UX/UI solution. The diagrams must cover **all** interactions as detailed in the scenario. When creating the use case diagrams, students **must** use a standard and consistent approach for symbols and notation.
Task 2	• Ideally, students will design the UX/UI solution planned in Task 1. However, if students deviate from their plan(s) they should not be penalised when assessing Task 2. • To confirm assessment decisions made for some of the criteria for this task, the OCR assessor will need to be able to see the final high-fidelity prototype. Therefore students must, provide either: o The final high-fidelity prototype in a file format which allows it to be viewed without the need to install any specialist software. o Video/screen recordings of the final high-fidelity prototype being demonstrated.
P4	• Students **must** create diagrams which show the interaction flows and navigation routes through the UX/UI solution and how the screens will link together. Students **must** use at least **one** of the design tools in Topic Area 3.1.1. The diagrams **must** contain enough detail to enable them to be interpreted by someone who hasn't seen them before.
P5	• Students **must** create diagrams which show the steps to be completed and decisions to be made for each process required in the UX/UI solution. Students **must** use at least **one** of the design tools in Topic Area 3.1.1. The diagrams **must** contain enough detail for them to be interpreted by someone who hasn't seen them before.
P6	• Students **must** create diagrams which show how users will complete actions when using the UX/UI solution. Students **must** use at least **one** of the design tools from Topic Area 3.1.1. The diagrams **must**

	contain enough detail for them to be interpreted by someone who hasn't seen them before.
P7	• Students **must** create a high-fidelity protype for the UX/UI solution that meets the requirements detailed in the scenario. The high-fidelity prototype can be created in generic software applications or interface prototyping software. The final UX/UI solution high-fidelity prototype will be sufficient evidence for this assessment criterion.
P8	• Students **must** describe how the UX/UI solution will meet the needs of the users specified in the scenario. Topic Area 1.2 details application end use considerations which may be applicable to the scenario.
P9	• Students **must** check the appropriateness and suitability of the UX/UI solution using UI audit and interface metrics in Topic Area 3.2.
M3	• Students **must** explain how they have applied each of the navigation design principles in Topic Area 1.3 to the UX/UI solution.
M4	• Students **must** explain how Schneiderman's 8 Golden Rules of interface design in Topic Area 1.3 have informed the UX/UI solution. The criterion is achieved if students explain how at least **four** of Schneiderman's 8 Golden Rules have informed the UX/UI solution.
M5	• Students must justify the checking used in P9.
D2	• Students **must** add appropriate error handling and feedback to the high-fidelity prototype created in P7. The UX/UI solution **must** handle user errors and provide informative feedback enabling users to self-resolve issues. The final UX/UI solution high-fidelity prototype will be sufficient evidence for this assessment criterion. Where students do not achieve P7, it is still possible to achieve D2.
D3	• Students **must** assess the UX/UI solution in relation to the **three** design psychology principles in Topic Area 1.4. Students **could** also consider other psychology principles they have studied.
D4	• Students **must** assess the UX/UI solution in relation to the **four** UX/UI interface standards in Topic Area 1.6.
Task 3	• When creating the UX/UI showcase, students will need to decide on an appropriate showcase format. Topic Area 4.1 includes examples of showcase formats. However, this is not an exhaustive list and students could choose an alternative appropriate format. • To reduce assessment burden, centres could consider limiting the duration of students' UX/UI showcases. • Where relevant to the showcase format, centres and/or students could record the delivery of UX/UI showcases for the purpose of internal assessment. However, please note there is no requirement to submit audio or visual recordings of students delivering UX/UI showcases for moderation.
P10	• Students **must** create a UX/UI showcase appropriately designed for the client. Topic Area 4.1 includes showcase considerations relating to design. To confirm assessment decisions, the OCR assessor will need to consider the appropriateness of the UX/UI showcase design. Therefore, students **must** provide suitable evidence in the form of a slide deck, screenshots, photographs, screen recordings, presentation notes, a script, supporting visual stimuli, for example. There is no requirement to submit audio or visual recordings of students delivering UX/UI showcases.
P11	• Students **must** deliver a UX/UI showcase that communicates the UX/UI solution using content appropriate for the client. Students **must** use a showcase format that is appropriate for the client. Topic Area 4.1 includes showcase considerations relating to content. To confirm assessment decisions, the OCR assessor will need to consider the appropriateness of the UX/UI showcase content. Students **must**

	provide suitable evidence in the form of a slide deck, screenshots, photographs, screen recordings, presentation notes, a script, supporting visual stimuli, for example. There is no requirement to submit audio or visual recordings of students delivering UX/UI showcases.
M6	• Students **must** deliver the UX/UI showcase content using the techniques for effective communication in Topic Area 4.2. To confirm assessment decisions, the OCR assessor will need to consider students' use of techniques for effective communication. Centres **must** provide an individualised, teacher observation record form for each student to evidence they have met this criterion. Students must also read and sign the Teacher Observation Record form (ToR). The criterion is achieved if students demonstrate at least **three** of the techniques for effective communication.
P12	• Students **must** describe the strengths and weaknesses of the UX/UI solution in relation to the requirements identified in **Task 1** and the design principles in Topic Area 1.3.
M7	• Having considered the strengths and weaknesses of the UX/UI solution (P12), students **must** discuss how the UX/UI solution could be improved. Students **could** consider the potential improvements in Topic Area 5.2.
D5	• Students **must** evaluate the effectiveness of processes used to plan and design the UX/UI solution. Students **must** incorporate an assessment of the effectiveness of the tools and techniques used.

Synoptic assessment

Some of the knowledge, understanding and skills needed to complete this unit will draw on the learning in Unit/s F160 and F161.

This table details these synoptic links.

Unit F162: Designing and communicating UX/UI solutions		Unit F160: Fundamentals of application development	
Topic Area		Topic Area	
1	Principles of UX and UI design	1 5	Types of software used in application design Principles of human computer interaction (HCI)
2	Requirements and ideation of UX/UI solutions	2 3 4 5	Software development models Project planning Application design scoping Principles of human computer interaction (HCI)
3	Design UX/UI solutions	4 5	Application design scoping Principles of human computer interaction (HCI)
4	Communicate UX/UI solutions	1 5 6	Types of software used in application design Principles of human computer interaction (HCI) Communication skills required in application development
5	Review and improve UX/UI solutions	1 2 3 4 5	Types of software used in application design Software development models Project planning Application design scoping Principles of human computer interaction (HCI)

Unit F162: Designing and communicating UX/UI solutions		Unit F161: Developing application software	
Topic Area		Topic Area	
1	Principles of UX and UI design	1	Application software considerations
2	Requirements and ideation of UX/UI solutions	1 2 6	Application software considerations Data and flow in application software Legal considerations
3	Design UX/UI solutions	1 2 5	Application software considerations Data and flow in application software Operational considerations
4	Communicate UX/UI solutions	1 4 5 6	Application software considerations Data and flow in application software Operational considerations Legal considerations
5	Review and improve UX/UI solutions	1 2 4 5 6	Application software considerations Data and flow in application software Application software security Operational considerations Legal considerations

More information about synoptic assessment in these qualifications can be found in **Section 5.2 Synoptic assessment.**

4.3.2 Unit F163: Game development

Unit Aim

Game development is a large sector of the software industry. It's a market that's constantly growing, with thousands of games being released each year, covering many genres across numerous platforms. This unit will help you develop knowledge, understanding, and skills in designing and developing prototypes for game concepts. It will help you to consider the mechanics needed to make games work, as well as the visuals needed to build a game environment.

In this unit you will learn how types and genres of digital games and their characteristics affect game design. You will then learn how to plan, design, create, and test game prototypes. You will gain the technical skills to create game environments and game functionality and learn how to test game prototypes to ensure they function as intended.

Unit F163: Game development	
Topic Area 1: Game design	
Teaching content	**Exemplification**
1.1 Types and genres of digital games	
1.1.1 Types of game □ 2D □ 3D □ Immersive games (Augmented Reality, Virtual Reality, Mixed Reality) □ Massive Multiplayer Online (MMO) games □ Massive Multiplayer Online Role-Playing Games (MMORPG) □ Role Playing Games (RPG) □ Platform □ Simulation	To include: □ The features and characteristics of each game type □ The differences between each game type □ How game type impacts game development
1.1.2 Genres of game □ Action □ Educational □ Puzzle and trivia □ Quest □ Sports □ Strategy	To include: □ The features and characteristics of each game genre □ How game genre impacts game development
1.1.3 Gaming platforms □ Types of gaming platform	To include: □ The features and characteristics of gaming platforms □ Differences between gaming platforms □ How the features and characteristics of gaming platforms impact game development Examples of **gaming platforms** may include: □ Cross platform □ Gaming consoles □ Online streaming □ PC gaming □ Smart mobile gaming □ Television streaming □ Virtual Reality (VR) Does not include: □ Technical specifications of hardware needed for gaming

1.1.4 Pan European Game Information (PEGI) Certificates □ Age ratings □ Content descriptions	To include: □ The features and characteristics of games which meet each current PEGI rating □ How PEGI ratings impact game development
1.2 Principles of game design	
1.2.1 Game concept □ Game purpose □ Game audience □ Story □ Unique Selling Proposition (USP)	To include: □ How game purpose informs the game concept □ How game audience informs the game concept □ How story informs the game concept □ What makes the game concept appealing, engaging and marketable □ How game concepts impact game development
1.2.2 Game and gameplay elements □ Game elements • Goals/objectives • Aesthetics/visuals • Game world dimensions ○ Environmental ○ Physical ○ Temporal ○ Emotional ○ Ethical • Theme and story □ Gameplay elements • Competition • Outcome and feedback • Player immersion • Player interaction • Progression • Reward/accomplishment • Scoring • Strategy and chance	To include: □ The purpose and use of each game and gameplay element □ The features and characteristics of each game and gameplay element □ How each game and gameplay element is used effectively in game development □ How game and gameplay elements are used in combination to maintain player interest and engagement
1.2.3 Game assets □ Animation □ Backgrounds □ Main characters □ Non-Player Characters (NPCs) □ Objects □ Scenery □ Sounds □ Textures □ Video	To include: □ The purpose and use of each game asset type □ The features and characteristics of each game asset type □ How game assets are used effectively in game development □ How the personification of game assets can be used to make gameplay more realistic □ How game assets are used in combination to maintain player interest and engagement
1.2.4 Game mechanics □ Character and object movement □ Character and object navigation □ Game actions and events □ Game play controls □ Game start mechanisms □ Game end mechanisms □ Inputs and outputs □ Collision detection □ Player interaction and feedback	To include: □ The purpose and use of each game mechanic type □ The features and characteristics of each game mechanic type □ How game mechanics are used effectively in game development □ How game mechanics are used to make gameplay more realistic

□ Scoring and timing mechanisms □ Shortcuts and cheats	□ How game mechanics are used in combination to maintain player interest and engagement
Topic Area 2: Plan and design high-fidelity game prototypes	
Teaching content	**Exemplification**
2.1 Tools to plan and design game prototypes	
2.1.1 Game Design Documents (GDDs) □ Format, layout and templates of GDDs □ Content of GDDs • Client requirements • Executive summary of game concept • Success criteria • Game and gameplay elements • Game assets • Game mechanics	To include: □ The purpose of, and audiences for, GDDs □ How the format, layout and structure of GDDs impact their effectiveness □ How to create GDDs for game prototypes
2.1.2 Game planning and design tools □ Tools to document designs for game visuals • Concept art • Storyboard • Assets list □ Tools to document plans for game mechanics • Decision trees • Flowchart • Pseudo code	To include: □ The purpose and use of each planning and design tool □ The format, layout and structure of each planning and design tool □ How to use tools to document plans and designs for game prototypes Does not include: □ Using project management tools or documentation
Topic Area 3: Create high-fidelity game prototypes	
Teaching content	**Exemplification**
3.1 Tools and techniques to source and prepare assets	
□ Sources of assets • Internet • Stock libraries □ Preparation of assets □ File formats and properties used in game creation □ Asset naming conventions used in game creation	To include: □ How to use internet and stock libraries to search for suitable assets for use in game protypes □ How to prepare assets for use in game prototypes □ How to select asset file formats and properties □ How to use naming conventions so assets are identifiable Examples of techniques to **prepare assets** may include: □ Sizing □ Changing resolution □ Cropping □ Changing length/duration of sound, video or animation □ Animating static images to create moving objects or characters □ Duplicating graphics to create larger backgrounds or textures □ Creating different versions of character assets to personify movement Does not include: □ Creating original assets

3.2 Technical skills to create game environments and game functionality	
□ Game engine tools • Asset management • Object controls • Animation systems • Physics engine/collision detection and response • Rendering engine • Sound support • Scripting environment • Libraries □ Programming techniques • Variables, constants, operators, inputs, outputs and assignments • Sequence, selection and iteration • Conditions using comparison, arithmetic and Boolean operators • File handling • Sub programs (sub routines/functions/procedures)	To include: □ How to use game engine tools/programming techniques to create game scenes/rooms/environments □ How to use game engine tools/programming techniques to implement game functionality Does not include: □ Writing code in a specific programming language to develop game prototypes
Topic Area 4: Test high-fidelity game prototypes	
Teaching content	**Exemplification**
4.1 Game prototype testing	
□ Testing methods • Dry run • Iterative • Test plan • Trace tables □ Testing types • Functionality testing • Performance testing • Play testing • Compatibility testing □ Elements of game prototypes to test • Actions and events • Audio effects • Character movement and navigation • Consistency of graphics • Game play controls • Game progression/levels • Player interaction and feedback • Scoring and timing mechanisms • Usability and gaming experience • User interface and functionality □ Results analysis and remedial action	To include: □ The structure, content and use of testing methods □ How and why to test iteratively both during prototype creation and post-prototype creation □ The purpose of each testing type □ The features and characteristics of each testing type □ When it is appropriate to use each testing type □ How to plan testing to ensure game prototypes function as intended □ How to implement testing to ensure game prototypes function as intended □ How to analyse testing results and identify remedial action Does not include: □ Implementing remedial action

Topic Area 5: Review and improve high-fidelity game prototypes	
Teaching content	**Exemplification**
5.1 Techniques to review the fitness for purpose of game prototypes	
□ Suitability for meeting: • Client requirements • Planning and design requirements □ Audio-visual/aesthetics quality □ Game and gameplay elements □ Player interaction and engagement □ Player suitability/appeal	To include: □ How to assess strengths and weaknesses of game prototypes □ How to compare game prototypes against requirements □ How to assess the quality and appropriateness of audio-visual/aesthetics □ How to assess the appropriateness and effectiveness of game and gameplay elements used □ How to assess the appropriateness and effectiveness of player interaction and engagement □ How to assess the appropriateness and effectiveness of player suitability/appeal
5.2 Improvements to, and further developments for, game prototypes	
5.2.1 Improvements □ Audio □ Gameplay □ Graphics □ Levels and progression □ Lifelikeness □ Video/animation	To include: □ How to assess potential improvements to game prototypes Does not include: □ Implementing improvements to game prototypes
5.2.2 Further development opportunities □ Building gaming communities □ Facilitating in-game purchases □ Marketing opportunities □ Release to gaming platforms □ Widen scope of game concept	To include: □ How to assess potential further development opportunities for game prototypes Does not include: □ Implementing further development recommendations to game prototypes

Assessment criteria

Section 6.4 provides full information on how to assess the NEA units and apply the assessment criteria.

These are the assessment criteria for the tasks for this unit. The assessment criteria indicate what is required in each task. Students' work must show that all aspects of a criterion have been met in sufficient detail for it to be **successfully achieved** (see **Section 6.4.1**). If a student's work does not fully meet a criterion, you must not award that criterion.

The command words used in the assessment criteria are defined in **Appendix B**.

<table>
<tr><th>Pass</th><th>Merit</th><th>Distinction</th></tr>
<tr><td>P1: Describe a concept for a game prototype that meets the client requirements as detailed in the scenario.</td><td></td><td rowspan="3">D1: Discuss how the planned game and gameplay elements maintain player interest and engagement in the game prototype.</td></tr>
<tr><td>P2: Identify the assets required for the game prototype.</td><td rowspan="2">M1: Explain how assets are used in combination in the game prototype.</td></tr>
<tr><td>P3: Design game visuals appropriate for the game prototype.</td></tr>
<tr><td>P4: Describe the game mechanics to be used in the game prototype.</td><td>M2: Plan game mechanics appropriate for the game prototype using game planning tools.</td><td>D2: Discuss how the planned game mechanics are used in combination to maintain player interest and engagement in the game prototype.</td></tr>
<tr><td>P5: Describe how the game prototype will be tested.</td><td>M3: Justify the appropriateness of the testing.</td><td></td></tr>
<tr><td>P6: Source assets appropriate for use in the game prototype.</td><td rowspan="2">M4: Prepare assets appropriately for use in the game prototype.</td><td rowspan="2"></td></tr>
<tr><td>P7: Create an appropriate game environment using game engine tools.</td></tr>
<tr><td>P8: Implement character and/or object movement and navigation appropriate for the game prototype.</td><td>M5: Implement collision detection appropriate for the game prototype.</td><td>D3: Create a cohesive game prototype combining game environment, assets and mechanics.</td></tr>
<tr><td>P9: Implement game play controls appropriate for the game prototype.</td><td rowspan="2">M6: Implement scoring and timing mechanisms appropriate for the game prototype.</td><td rowspan="2">D4: Implement player interaction and feedback appropriate for the game prototype.</td></tr>
<tr><td>P10: Implement game start and end mechanisms appropriate for the game prototype.</td></tr>
<tr><td>P11: Test the game prototype and document results.</td><td>M7: Analyse test results documenting any required remedial action.</td><td rowspan="2">D5: Discuss potential improvements and further development opportunities for the game prototype.</td></tr>
<tr><td>P12: Assess the suitability of the game prototype for meeting the requirements.</td><td></td></tr>
</table>

Assessment guidance

This assessment guidance gives you information relating to the assessment criteria. There might not be additional assessment guidance for each assessment criterion. It is included only where it is needed.

Assessment Criteria	Assessment guidance
P1	• Students **must** describe a concept for a game prototype which meets **all** the client requirements. The description of the concept **must** include the content in Topic Area 1.2.1.
P2	• Students **must** identify the essential assets required for the game prototype. To achieve this criterion, students are not required to identify all non-essential assets. This assessment criterion **could** be evidenced in an assets list and/or via the design(s) for the game visuals (P3).
P3	• Students **must** design the game visuals for the game prototype using at least one of the game design tools in Topic Area 2.1.2. The design(s) **must** contain enough detail for them to be interpreted by someone who hasn't seen them before.
P4	• Students **must** describe the game mechanics that they will use in the game prototype that are essential to the gameplay.
P5	• Students **must** describe the testing methods and testing types they will use to test the game prototype and the elements of the game prototype they intend to test. The description of how the game prototype will be tested **could** include the content in Topic Area 4.1.
M1	• Students **must** explain how the essential assets identified in P2 will be used in the game prototype. Students **must** explain which of the essential assets will be static and which will work in combination. This assessment criterion **could** be evidenced in an assets list and/or via annotations on the design(s) for the game visuals.
M2	• Students **must** plan **all** the game mechanics detailed in P4 using at least **one** of the game planning tools in Topic Area 2.1.2. Where students do not achieve P4, it is still possible to achieve M2. The planning **must** contain enough detail for it to be interpreted by someone who hasn't seen them before.
M3	• Students **must** justify the approach to testing detailed in P5.
D1	• Students **could** include the content in Topic Areas 1.2.2 and 1.2.3 in the discussion.
D2	• Students **could** include the content in Topic Area 1.2.4 in the discussion.
Task 2	• Ideally students will create the game prototype planned and designed in **Task 1**. However, if students deviate from the plan(s) and/or design(s) they should not be penalised when assessing **Task 2**. • To confirm assessment decisions made for this task, the OCR assessor will need to be able to see the final game prototype. Therefore, students **must,** provide either: ○ The final game prototype in a format which allows it to be played without the need to install any specialist software and instructions on how to play the game.

	○ Video/screen recordings of the final game prototype being demonstrated. This is especially useful if the skill level required to play the game is high.
P6	• Students **must** source assets for use in the game prototype that are appropriate for the game concept detailed in P1. The final game prototype will be sufficient evidence for this assessment criterion.
P7	• Using assets sourced in P6, students **must** create a game environment (screens/rooms/levels/stage), appropriate for the game concept detailed in P1. Students **must** use the game engine tools in Topic Area 3.2. The final game prototype will be sufficient evidence for this assessment criterion.
P8	• Students **must** add character and/or object movement and navigation to the game prototype, so the game functions as intended. Students **must** use the game engine tools and/or programming techniques in Topic Area 3.2. The final game prototype will be sufficient evidence for this assessment criterion.
P9	• Students **must** add game play controls to the game prototype, so the game functions as intended. Students **must** use the game engine tools and/or programming techniques in Topic Area 3.2. The final game prototype will be sufficient evidence for this assessment criterion.
P10	• Students **must** add game start and end mechanisms to the game prototype, so the game functions as intended. Students **must** use the game engine tools and/or programming techniques in Topic Area 3.2. The final game prototype will be sufficient evidence for this assessment criterion.
M4	• Students **must** prepare **all** assets sourced in P6, so they are appropriate for use in the game prototype. Topic Area 3.1 has examples of techniques students could use to prepare assets**.** Students **could** prepare assets in the game engine software or in external graphic software. The final game prototype will be sufficient evidence for this assessment criterion.
M5	• Students **must** add collision detection to the game prototype, so the game functions as intended. Students **must** use the game engine tools and/or programming techniques in Topic Area 3.2. The final game prototype will be sufficient evidence for this assessment criterion.
M6	• Students **must** add scoring and timing mechanisms to the game prototype, so the game functions as intended. Students **must** use the game engine tools and/or programming techniques in Topic Area 3.2. The final game prototype will be sufficient evidence for this assessment criterion.
D3	• Students **must** create a game prototype where all components of the game work, and fit, together. The final game prototype will be sufficient evidence for this assessment criterion.
D4	• Students **must** add player interaction and feedback to the game prototype, so the game functions as intended. Students **must** use the game engine tools and/or programming techniques in Topic Area 3.2. The final game prototype will be sufficient evidence for this assessment criterion.
P11	• Students **must** test the game prototype and document results. Ideally students will use the approach described and justified in **Task 1**. However, if students deviate from the proposed testing they should not be penalised. • Students **must** have evidence of the actual test results. For example, screen shots, photographs or video/screen recordings.

P12	• Students **must** assess the suitability of the game prototype for meeting the requirements in Topic Area 5.1.
M7	• Students **must** analyse the test results generated in P11 and explain any remedial action required to resolve the issues found during testing. Students are not expected to fix errors found in the game prototype during final testing.
D5	• Having assessed the suitability of the game prototype (P12) and analysed test results (M7), students **must** discuss potential improvements and further developments to the game prototype.

Synoptic assessment

Some of the knowledge, understanding and skills needed to complete this unit will draw on the learning in Units F160 and F161.

This table details these synoptic links.

Unit F163: Game development		Unit F161: Developing application software	
Topic Area		Topic Area	
1	Game design concepts	1 6	Application software considerations Legal considerations
2	Plan and design high-fidelity game prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations
3	Create high-fidelity game prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations
4	Test high-fidelity game prototypes	1 2 3 4 5	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations
5	Review and improve high-fidelity game prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations

More information about synoptic assessment in these qualifications can be found in **Section 5.2 Synoptic assessment.**

4.3.3 Unit F164: Website development

Unit Aim

Websites are one of the major communication technologies used globally for a range of purposes. The requirement for websites to be accessed from a wide range of devices provides challenges to be considered when developing websites. As reliance on internet technology is increasingly important to consumers and businesses, websites need to keep up with demands. This requires a more immersive experience, which delivers personalised and tailored content to the screen of individual consumers.

In this unit you will learn how to design and develop a prototype website. You will learn about website principles and the components of web pages. You will learn how to design a website that can be viewed on a range of devices, together with how to make sure a website complies with accessibility requirements and guidance. You will also learn how to make websites more visible in search engine results by utilising SEO techniques. You will develop your skills by creating, testing, and reviewing a website prototype using a range of tools and techniques.

Unit F164: Website development	
Topic Area 1: Fundamentals of website development	
Teaching content	**Exemplification**
1.1 Website principles	
□ Domain name/Uniform Resource Locator (URL) • Structure □ Platform and browser compliance □ Device compliance • Sizes • Types □ W3C compliance • Protocols and guidelines • Web Content Accessibility Guidelines (WCAG) □ Site structures • Index page • Site map □ Web 2.0 □ Web 3.0	To include: □ The features and characteristics of each website principle □ How each principle impacts website development
1.2 Purpose of websites	
□ Advertise/promote □ Educate □ Entertain □ Influence □ Inform □ Market □ Sell	To include: □ The purpose of websites □ How the content, layout and style are adapted to meet the purpose
1.3 Website types	
□ Interactive □ Multimedia □ Responsive □ Single page □ Static □ Dynamic □ Content Management Systems (CMS)	To include: □ The features and characteristics of each website type □ The differences between each website type □ How different types of website can be combined □ How websites are evolving to provide increased personalisation of experience

1.4 Webpage components and structure	
□ Semantic page components • Interface designs □ Hyper Text Markup Language 5 (HTML5) or later versions □ Cascading Style Sheets 3 (CSS3) or later versions □ Client-side scripting □ Navigational components • Hyperlinks • Hotspots • Navigation bar □ User interactions □ Forms □ Tags □ Responsive design features • Compatibility ○ Browser ○ Device • Fluid grids • Media queries/break points • Relative sizing □ Libraries/Frameworks • HTML based • CSS based • JavaScript based • Hypertext Pre-processor (PHP) based □ Animation techniques	To include: □ The purpose and role of each webpage component □ How each component is used/implemented in the creation of webpages
1.5 Search Engine Optimisation (SEO) techniques	
□ Crawling □ Indexing □ Keywords □ Metadata □ Mobile-friendly □ Ranking	To include: □ How search engines find websites □ How website performance in search engines is impacted by optimisation □ The purpose and use of each SEO technique □ How each SEO technique is used to improve a website's performance in search engines
Topic Area 2: Plan and design high-fidelity website prototypes	
Teaching content	**Exemplification**
2.1 Planning and design considerations	
□ Client requirements • Purpose • Type of website • Target audience • Content of website □ User requirements □ Navigation system □ Interactive components • Buttons • Media controls • User input fields • Rollovers • Hyperlinks • Hotspots □ Assets • Text	To include: □ How each consideration impacts website development □ How purpose impacts domain name choice and hosting requirements

• Sound • Images • Video/animation • Forms □ House style • Colours • Fonts • Styles ○ Images ○ Text □ Plugins □ Responsive design features □ Search Engine Optimisation (SEO) □ W3C compliance □ Hosting requirements • Cost • Location • Security • Domain name	
2.2 Tools to plan and design website prototypes	
□ Tools to document ideas for website prototypes • Mind maps • Mood boards □ Tools to document plans and designs for website prototypes • Site plans • Visualisation diagrams • Wireframes • Storyboards • Assets list • House style sheet	To include: □ The purpose and use of each tool □ The components and conventions of each tool □ When it is appropriate to use each tool □ How to use tools to document ideas for website prototypes □ How to use tools to document plans and designs for website prototypes including: • House style • Content • Page layout • Page linking • Navigation systems • Interface • Functionality
Topic Area 3: Create high-fidelity website prototypes	
Teaching content	**Exemplification**
3.1 Tools and techniques to create website structures	
□ Folder structure • Templates • Assets • Pages □ Site page structure □ Index page location	To include: □ How to structure folder systems for website prototypes □ How to identify index page location for website prototypes
3.2 Techniques to source and prepare assets	
□ Sources of assets • Internet • Stock libraries □ Preparation of assets □ File formats and properties used in website creation □ Asset naming conventions used in website creation	To include: □ How to use internet and stock libraries to search for suitable assets for use as website content □ How to prepare assets for use as website content □ How to select asset file formats and properties which are used for website protypes □ How to use naming conventions so assets are identifiable

	Examples of techniques to **prepare assets** may include: □ Sizing □ Changing resolution □ Cropping □ Removing backgrounds □ Changing length/duration of sound, video, or animation □ Animating static images to create moving objects or banners Does not include: □ Creating original assets
3.3 Technical skills to create website pages	
□ Web authoring software tools • Visual design environment • Scripting environment with coding assistance • Template creation • Cascading style sheets (CSS) ○ Box model ○ Website/page formatting ○ Content formatting • Responsive design features • Form controls • Interactive features and controls • Preview and publishing • Libraries/Frameworks • Search Engine Optimisation (SEO)	To include: □ How to use web authoring software tools to create editable templates for use in website prototypes □ How to use web authoring software tools to create individual webpages of website prototypes □ How to create website prototypes which meet current W3C and accessibility guidelines Examples of **software tools** use may include: □ Creating editable templates for individual webpages □ Implementing house style using CSS □ Using template pages to make sure the layout and style is consistent throughout website prototypes □ Creating working navigation systems for website prototypes □ Inserting appropriate content □ Creating forms in webpages to gain user response □ Creating responsive templates and webpages □ Using Libraries/Frameworks in webpages □ Applying SEO techniques to webpages of website prototypes
Topic Area 4: Test high-fidelity website prototypes	
Teaching content	**Exemplification**
4.1 Website prototype testing	
□ Testing methods • Dry run/trace table • Iterative • Test Plan □ Testing types • Technical testing • Viewpoint testing • User testing □ Elements of website prototypes to test • Content display • Ease of use • Hyperlinking	To include: □ The structure, content and use of testing methods □ How and why to test iteratively both during prototype creation and post-prototype creation □ The purpose of each testing type □ The features and characteristics of each testing type □ When it is appropriate to use each testing type

• Interactive elements • Multiple browser testing • Multiple device testing • Multiple viewpoint size testing • Navigation features • Pages display • Readability of content □ Results analysis and remedial action	□ How to plan testing to ensure website prototypes function as intended □ How to implement testing to ensure website prototypes function as intended □ How to analyse testing outcomes and identify remedial action Does not include: □ Implementing remedial action
Topic Area 5: Review and improve the effectiveness of high-fidelity website prototypes	
Teaching content	**Exemplification**
5.1 Techniques to review the effectiveness of website prototypes	
□ Suitability for meeting: • Client requirements • User requirements □ Accessibility □ Device independence/compatibility □ Responsive design □ Search Engine Optimisation (SEO) techniques used	To include: □ How to assess strengths and weaknesses of website prototypes □ How to compare website prototypes against requirements □ How to assess the device accessibility, compatibility and responsiveness of website prototypes □ How to assess the effectiveness of SEO techniques used in website prototypes
5.2 Improvements to, and further developments for, website prototypes	
5.2.1 Constraints and improvements □ Constraints • Legislation • Libraries/Frameworks • Skills • Software • Time □ Improvements • Accessibility • Browser independence/compatibility • Content, visuals and interaction • Device independence/compatibility • Domain name • Search Engine Optimisation (SEO) • Security	To include: □ How to assess the impact of constraints on website prototypes □ How to assess potential improvements to website prototypes Does not include: □ Implementing improvements to website prototypes
5.2.2 Further development opportunities □ Extra content/features □ Further user interactivity □ Hosting considerations □ Payment gateways/processors	To include: □ How to assess potential further development opportunities for website prototypes Does not include: □ Implementing further development opportunities to website prototypes

Assessment criteria

Section 6.4 provides full information on how to assess the NEA units and apply the assessment criteria.

These are the assessment criteria for the tasks for this unit. The assessment criteria indicate what is required in each task. Students' work must show that all aspects of a criterion have been met in sufficient detail for it to be **successfully achieved** (see **Section 6.4.1**). If a student's work does not fully meet a criterion, you must not award that criterion.

The command words used in the assessment criteria are defined in **Appendix B**.

<table>
<tr><th>Pass</th><th>Merit</th><th>Distinction</th></tr>
<tr><td>P1: Describe the client and user requirements for the website prototype.</td><td rowspan="2">M1: Explain the Libraries/ Frameworks required for the website prototype development.</td><td rowspan="2">D1: Justify the Search Engine Optimisation (SEO) techniques to be used in the website prototype.</td></tr>
<tr><td>P2: Explain the hosting requirements for the website prototype.</td></tr>
<tr><td>P3: Create a design of the website structure, navigation system and a content overview.</td><td rowspan="2">M2: Explain how the house style for the website prototype is appropriate for the client requirements.</td><td rowspan="2">D2: Assess website prototype design choices in relation to W3C and accessibility compliance.</td></tr>
<tr><td>P4: Create a design of the webpage template(s) to show the page layout and the house style.</td></tr>
<tr><td>P5: Identify assets required for the website prototype.</td><td></td><td></td></tr>
<tr><td>P6: Describe how the website prototype will be tested.</td><td>M3: Justify the appropriateness of the testing.</td><td></td></tr>
<tr><td>P7: Create an appropriate website structure for the website prototype.</td><td></td><td></td></tr>
<tr><td>P8: Prepare assets appropriate for use as components in the website prototype.</td><td></td><td></td></tr>
<tr><td rowspan="2">P9: Create the interactive and navigational components appropriate for the website prototype.</td><td>M4: Implement W3C and accessibility compliance in the website prototype.</td><td rowspan="2">D3: Implement appropriate Search Engine Optimisation (SEO) techniques in the website prototype.</td></tr>
<tr><td rowspan="2">M5: Implement appropriate responsive design features in the website prototype.</td></tr>
<tr><td rowspan="2">P10: Create the website prototype using web authoring software tools.</td><td rowspan="2">D4: Use appropriate Libraries/Frameworks to create the website prototype.</td></tr>
<tr><td>M6: Use Cascading Style Sheets (CSS) to implement an appropriate and consistent style in the website prototype.</td></tr>
<tr><td>P11: Test the website prototype and document results.</td><td>M7: Analyse test results documenting any required remedial action.</td><td rowspan="2">D5: Discuss potential improvements and further development opportunities for the website prototype.</td></tr>
<tr><td>P12: Assess the suitability of the website prototype for meeting the requirements.</td><td></td></tr>
</table>

Assessment guidance

This assessment guidance gives you information relating to the assessment criteria. There might not be additional assessment guidance for each assessment criterion. It is included only where it is needed.

Assessment Criteria	Assessment guidance
P1	• Students **must** describe both the client **and** user requirements. Students **must** expand the description into specific requirements which can be used as criteria to review against in **Task 3**.
P2	• Students **must** explain the hosting requirements for the website prototype. The explanation of the hosting requirements **must** include the content in Topic Area 2.1.
P3	• Students **must** create a design of the website structure for the proposed website prototype. Students **could** use the website planning and design tools in Topic Area 2.2. The site plan **must** include the navigation system and a summary of individual page content. The site plan **must** contain enough detail for it to be interpreted by someone who hasn't seen it before.
P4	• Students **must** create a design for the webpage template(s). Students **could** use the website planning and design tools in Topic Area 2.2. The design **must** show how the pages will be laid out. Through the design, students **must** also define the house style for the website prototype. The designs **must** contain enough detail for them to be interpreted by someone who hasn't seen them before.
P5	• Students **must** identify the essential assets required for the website prototype. To achieve this criterion, students are not required to identify all non-essential assets. This assessment criterion could be evidenced in an assets list.
P6	• Students **must** describe the testing methods and testing types they will use to test the website prototype and the elements of the website prototype they intend to test. The description of how the software solution will be tested **could** include the content in Topic Area 4.1.
M1	• Students **must** explain which Libraires/Frameworks they are using to develop the website prototype. The explanations **must** include how the chosen Libraries/Frameworks will allow the website prototype to function as intended.
M2	• Students **must** explain how the house style(s) defined in P4 is appropriate for the client as detailed in the scenario.
M3	• Students **must** justify the approach to testing detailed in P6.
D1	• Students **must** explain how the search engine techniques they intend to use will ensure the website prototype is visible to search engines.
D2	• Students **must** produce an assessment of how the design choices for the website prototype meet current W3C and accessibility guidelines.
Task 2	• Ideally students will create the website prototype planned and designed in Task 1. However, if students deviate from the plan(s) and/or design(s) they should not be penalised when assessing Task 2. • To confirm assessment decisions made for this task, the OCR assessor will need to be able to see the final website prototype. Therefore, students must, provide either: o The final website prototype in a format which allows it to be viewed/used without the need to install any specialist software. o Video/screen recordings of the final website prototype being demonstrated.
P7	• Students **must** create a website structure including the folder structure, site pages structure and index page location. This

	assessment criterion **could** be evidenced in screen shots or photographs showing the website folders and files.
P8	• Students **must** prepare assets sourced for use as components in the website prototype. Topic Area 3.1 has examples of techniques students could use to prepare assets**.** The final website prototype will be sufficient evidence for this assessment criterion.
P9	• Students **must** create interactive and navigational components to enable the website to function as intended. Students **could** use the web authoring software tools in Topic Area 3.3. The final website prototype will be sufficient evidence for this assessment criterion.
P10	• Students **must** create the prototype website using the web authoring tools in Topic Area 3.2. The final website prototype will be sufficient evidence for this assessment criterion.
M4	• Students **must** add W3C and accessibility compliance to the website prototype using the web authoring software tools in Topic Area 3.3. The final website prototype will be sufficient evidence for this assessment criterion.
M5	• Students **must** add appropriate responsive design features to the website prototype using the web authoring software tools in Topic Area 3.3. Students **must** ensure the website prototype functions as intended on the devices detailed in the scenario. This assessment criterion **could** be evidenced in videos or photographs showing the website prototype working as intended on the devices detailed in the scenario
M6	• Students **must** use CSS to implement a consistent style in the website prototype using the web authoring software tools in Topic Area 3.3. The style must be appropriate for the requirements detailed in the scenario. Students **must** include evidence of the CSS styles implemented. This assessment criterion **could** be evidenced in screen shots, photographs or video/screen recordings showing the CSS styles implemented.
D3	• Students **must** add Search Engine Optimisation (SEO) techniques to the website prototype using the web authoring tools in Topic Area 3.3. The techniques used **must** enable the website prototype to be visible to search engines. This assessment criterion **could** be evidenced in screen shots, photographs or video/screen recordings showing the techniques used.
D4	• Students **must** make use of Libraries/Frameworks to meet the requirements detailed in the scenario. Students **must** include evidence of the Libraries/Frameworks they have used. This assessment criteria **could** be evidenced in screen shots, photographs or video/screen recordings showing the Libraries/Frameworks used.
P11	• Students **must** test the website prototype and document results. Ideally students will use the approach described and justified in **Task 1**. However, if students deviate from the proposed testing they should not be penalised. • Students **must** also have evidence of the actual test results. For example screen shots, photographs or video/screen recordings.
P12	• Students **must** assess the suitability of the website prototype for meeting the requirements in Topic Area 5.1.
M7	• Students **must** analyse the test results generated in P11 and explain any remedial action required to resolve the issues found during testing. Students are not expected to fix errors found in the website prototype during final testing.
D5	• Having assessed the suitability of the website prototype (P12) and analysed test results (M7), students **must** discuss potential improvements and further developments to the website prototype.

Synoptic assessment

Some of the knowledge, understanding and skills needed to complete this unit will draw on the learning in Unit/s F160 and F161.

This table details these synoptic links.

Unit F164: Website development		Unit F160: Fundamentals of application development	
Topic Area		Topic Area	
1	Fundamentals of website development	1 5	Types of software used in application design Human computer interface and interaction
2	Plan and design high-fidelity website prototypes	2 3 4 5	Software development models Planning application developments Application design scoping Human computer interface and interaction
3	Create high-fidelity website prototypes	2 5	Software development models Human computer interface and interaction
4	Test high-fidelity website prototypes	2 5	Software development models Human computer interface and interaction
5	Review and improve the effectiveness of high-fidelity website prototypes	1 2 3 4 5	Types of software used in application design Software development models Planning application developments Application design scoping Human computer interface and interaction

Unit F164: Website development		Unit F161: Developing application software	
Topic Area		Topic Area	
1	Fundamentals of website development	1 2 3	Application software considerations Data and flow in application software API and protocols
2	Plan and design high-fidelity website prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations
3	Create high-fidelity website prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations
4	Test high-fidelity website prototypes	1 2 3 4 5	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations
5	Review and improve the effectiveness of high-fidelity website prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations

More information about synoptic assessment in these qualifications can be found in **Section 5.2 Synoptic assessment.**

4.3.4 Unit F165: Immersive technology solution development

Unit Aim

Immersive technologies merge the physical world with digital or simulated reality in a way that creates unique user experiences with natural engagement. The 360° space within immersive experiences allows users to look at and see content in any direction so virtual elements of the environment are accepted as real. Immersive technologies include Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR). AR blends digitally generated information onto the user's real environment. VR uses digitally generated information to provide a complete sense of immersion. MR is an interactive depiction/view of a combination of real world and digitally generated information and the use of virtual objects that can interact with the environment.

In this unit you will learn the principles of immersive technologies including the types, how each type can be used, the current advantages and disadvantages, and supporting hardware/technology. You will also learn the tools and techniques to plan, design and create immersive technology solution prototypes and how to test them to make sure they function as intended. Finally, you will learn how to review immersive technology solution prototypes and recommend how they could be improved and further developed.

Unit F165: Immersive technology solution development	
Topic Area 1: Principles of immersive technology	
Teaching content	**Exemplification**
1.1 Types and uses of immersive technology	
□ Types of immersive technology • Augmented Reality (AR) • Virtual Reality (VR) • Mixed Reality (MR) □ Immersive technology use	To include: □ The features of immersive technologies □ The differences between each immersive technology □ The advantages and disadvantages of each immersive technology □ How AR, VR, and MR are used within different sectors □ How the form and structure of immersive technology solutions are affected by purpose and use □ How immersive technology can be used to enhance the user experience within a realistic environment Examples of **sectors** using immersive technologies may include: □ Agriculture □ Architecture □ Education □ Entertainment, leisure and the media □ Health care and surgery □ Manufacturing □ Military □ Retail □ Sport Examples of **immersive technology use** may include: □ Concept visualisation □ Entertainment □ Maintenance □ Marketing/advertising □ Order fulfilment

	□ Simulations □ Training □ Virtual tours
1.2 Immersive technology concepts	
1.2.1 Augmented Reality (AR) □ AR types • Marker-based/object recognition • Markerless • Location-based • Superimposed □ Components of AR • Lenses • Processing • Sensing □ User interaction/layers • Static • Interactive □ Devices • AR glasses • Laptop/PC • Mobile devices • Smart devices	To include: □ How each AR type and component are used within AR solutions □ How users interact with AR solutions □ The features of devices which allow access to AR solutions □ How the features of devices impact AR design □ How AR can be used to enhance the user experience
1.2.2 Virtual Reality (VR) □ VR type • Non-immersive • Semi-immersive • Fully immersive □ Characteristics of VR • Virtual world • Immersive • Sensory feedback • Interactivity □ User interaction • Tracking sensors • Hand controllers • Audio □ Devices • Laptop/PC • Smart devices • VR glasses • VR headset and hand-held joystick	To include: □ How each VR type and characteristic are used within VR solutions □ How users interact with VR solutions □ The features of devices which allow access to VR solutions □ How the features of devices impact VR design □ How VR can be used to enhance the user experience
1.2.3 Mixed Reality (MR) □ MR concepts • Blend of physical and digital world • Unlocking interactions □ User interaction • Static • Interactive ○ Tracking sensors ○ Hand controllers ○ Audio □ Devices • Laptop/PC • Mobile devices • MR glasses/lenses • MR wearables • Smart devices	To include: □ How concepts of MR are used within solutions □ How users interact with MR solutions □ The features of displays to access MR solutions □ How the features of displays impact MR design □ How MR can be used to enhance the user experience □ How an immersive environment can have non-immersive and partially immersive aspects

□ Displays • Head mounted display (HMD) showing video • Immersive audio visual (AV) with 3D graphics with superimposed video on a monitor • Monitor-based video displays • Optical see-through Head Mounted Displays (HMDs)	Examples of **unlocking interactions** may include: □ Eye gaze □ Hand gestures/movements □ Head movement □ Physical handling
1.2.4 Technologies which support AR, VR and MR □ 3D modelling/scanning □ Multimedia □ Real-time tracking and registration □ Sensors	To include: □ The features and characteristics of each supporting technology □ How each supporting technology is used in immersive technology solutions □ How each supporting technology impacts immersive technology development Examples of **sensors** may include: □ Motion □ Proximity □ Image □ Gyroscope □ Accelerometers
Topic Area 2: Plan and design high-fidelity immersive technology solution prototypes	
Teaching content	**Exemplification**
2.1 Planning and design considerations	
□ Prototype planning considerations • Client requirements ○ Purpose ○ Target audience ○ Type of immersive technology • User requirements • Technical requirements ○ Hardware requirements ○ Devices required to access immersive technology prototype ○ Software considerations □ Prototype design considerations • Layout • Content ○ Assets • Triggers ○ Marker-based/object recognition ○ Markerless ○ Location/geo-based • Layers ○ Single layer ○ Multiple layers • User Interaction ○ Action flow ○ Static ○ Interactive • Degrees of freedom ○ Rotational movements - Pitch - Roll - Yaw ○ Translational movements	To include: □ How each consideration impacts immersive technology prototype planning □ How the hardware and devices available to users impacts immersive technology prototype planning □ How to identify immersive technology prototype requirements and success criteria □ How to identify immersive technology prototype content and assets □ How to design immersive technology prototypes which make use of triggers, layers, user interaction, degrees of freedom and fields of view □ Why frames per second and latency need to be considered when designing immersive technology prototypes Examples of **assets** may include: □ Audio □ External links □ Images □ Charts and graphs □ Text □ Videos Examples of **interactive user interactions** may include: □ Swipe □ Click/select □ Voice

- Left and right - Forwards and backwards - Up and down • Field of view • Frames per second • Latency	
2.2 Tools to plan and design immersive technology prototypes	
□ Tools to document ideas for immersive technology prototypes • Mind maps • Mood boards □ Tools to document plans and designs for immersive technology prototypes • Storyboards • Visualisation diagrams • Wireframes • Assets list • Flowcharts	To include: □ The purpose and use of each tool □ The components and conventions of each tool □ When it is appropriate to use each tool □ How to use tools to document ideas for immersive technology prototypes □ How to use tools to document plans and designs for immersive technology prototypes
Topic Area 3: Create high-fidelity immersive technology solution prototypes	
Teaching content	**Exemplification**
3.1 Techniques to source and prepare assets	
□ Sources of assets □ Preparation of assets □ File formats and properties □ Asset naming conventions used in immersive technology prototype creation	To include: □ What makes a good asset □ How to identify and select suitable assets for use in immersive technology prototypes □ How to prepare assets for use in immersive technology prototypes □ How to select/adapt file formats and properties for assets □ How to use naming conventions so assets are identifiable Examples of **sources of assets** may include: □ Books □ Internet □ Photographs □ Physical objects □ Stock libraries Examples of techniques to **prepare assets** may include: □ Sizing □ Changing resolution □ Cropping □ Removing backgrounds □ Changing length/duration of sound, video, or animation clips □ Selecting and adapting/repurposing 2D/3D assets Does not include: □ Creating original assets

3.2 Software features and techniques to create immersive technology prototypes	
□ Software features and techniques • Action/behaviour controls • Asset management • Drag/drop object manipulation • Environment lighting filters • Intelligent interaction controls • Layers/overlays management • Trigger controls	To include: □ How to use software features and techniques to implement immersive technology prototype design considerations □ How to use software features and techniques to create immersive technology prototypes □ How to use software features and techniques to implement supporting technologies in immersive technology prototypes Examples of **software features and technique** use may include: □ Creating motion and degrees of freedom □ Creating draggable and droppable objects □ Using effects to simulate lighting to emphasise the environment being represented □ Creating layers/overlays and related actions □ Creating triggers which contain graphical elements and shapes
Topic Area 4: Test high-fidelity immersive technology prototypes	
Teaching content	**Exemplification**
4.1 Immersive technology prototype testing	
□ Testing methods • Dry run/trace table • Iterative • Test plan □ Testing types • Functionality • Usability • Accessibility • Hardware • Immersiveness • Security • Compatibility □ Elements of immersive technology prototypes to test • Triggers • Layers • Interactions • Tracking • Degrees of freedom • Immersiveness • Battery consumption □ Results analysis and remedial action	To include: □ The structure, content and use of testing methods □ How and why to test iteratively both during prototype creation and post-prototype creation □ The purpose of each testing type □ The features and characteristics of each testing type □ When it is appropriate to use each testing type □ How to plan testing to ensure immersive technology prototypes function as intended □ How to implement testing to ensure immersive technology prototypes function as intended □ How to analyse testing results and identify remedial action Does not include: □ Implementing remedial action
Topic Area 5: Review and improve the effectiveness of high-fidelity immersive technology prototypes	
Teaching content	**Exemplification**
5.1 Techniques to review the effectiveness of immersive technology prototypes	
□ Suitability for meeting: • Client requirements • User requirements • Planning and design requirements	To include: □ How to assess strengths and weaknesses of immersive technology prototypes

□ Usability and immersiveness □ User experience and engagement	□ How to compare immersive technology prototypes against requirements □ How to assess the usability and immersiveness of immersive technology prototypes □ How to assess the user experience and engagement
5.2 Improvements to, and further developments for, immersive technology prototypes	
5.2.1 Improvements □ Functionality □ Usability □ Accessibility □ Hardware □ Immersiveness □ Security □ Compatibility □ Extra features	To include: □ How to assess potential improvements to immersive technology prototypes Does not include: □ Implementing improvements to immersive technology prototypes
5.2.2 Further development opportunities □ Availability of different resources/techniques □ Re-purposing	To include: □ How to assess potential further development opportunities for immersive technology prototypes Does not include: □ Implementing further development recommendations to immersive technology prototypes

Assessment criteria

Section 6.4 provides full information on how to assess the NEA units and apply the assessment criteria.

These are the assessment criteria for the tasks for this unit. The assessment criteria indicate what is required in each task. Students' work must show that all aspects of a criterion have been met in sufficient detail for it to be **successfully achieved** (see **Section 6.4.1**). If a student's work does not fully meet a criterion, you must not award that criterion.

The command words used in the assessment criteria are defined in **Appendix B**.

Pass	Merit	Distinction
P1: Describe the client and user requirements for the immersive technology prototype.	**M1: Explain** how the technical requirements meet the client and user requirements.	**D1: Justify** how the design for the immersive technology prototype meet the client and user requirements.
P2: Document appropriate ideas for the immersive technology prototype.		
P3: Design the layout for the immersive technology prototype.	**M2: Plan** how users will interact with the immersive technology prototype.	**D2: Discuss** how the immersive technology prototype enhances the user experience.
P4: Identify the assets required for the immersive technology prototype.	**M3: Explain** how assets will be used in combination in the immersive technology prototype.	
P5: Describe the functionality of the immersive technology prototype.		
P6: Describe how the immersive technology prototype will be tested.	**M4: Justify** the appropriateness of the testing.	

Pass	Merit	Distinction
P7: **Source** assets appropriate for use in the immersive technology prototype.	**M5: Prepare** assets appropriately for use in the immersive technology prototype.	
P8: Implement layers appropriate for the immersive technology prototype.	**M6: Use** software features and techniques to combine content appropriately in the immersive technology prototype.	**D3: Create** a cohesive immersive technology prototype combining environment, assets and actions.
P9: **Implement** triggers appropriate for the immersive technology prototype.		
P10: **Implement** user interaction appropriate for the immersive technology prototype.		**D4: Implement** supporting technologies appropriate for the immersive technology prototype.
P11: Test the immersive technology prototype and document results.	**M7**: **Analyse** test results documenting any required remedial action.	**D5: Discuss** potential improvements and further development opportunities.
P12: Assess the suitability of the immersive technology prototype for meeting the requirements.		

Assessment guidance

This assessment guidance gives you information relating to the assessment criteria. There might not be additional assessment guidance for each assessment criterion. It is included only where it is needed.

Assessment Criteria	Assessment guidance
P1	• Students **must** describe both the client **and** user requirements. Students **must** expand the description into specific requirements which can be used as criteria to review against in **Task 3**. This assessment criterion **could** be evidenced in a requirements specification.
P2	• Students **must** document ideas for the immersive technology prototype based on the requirements described in P1. Students **must** use at least **one** of the tools in Topic Area 2.2. Students **must** document at least **two** ideas on a mind map, for example.
P3	• Students **must** design the immersive technology prototype using the tools in Topic Area 2.2. The designs **must** clearly show the intended layout and contain enough detail for them to be interpreted by someone who hasn't seen them before.
P4	• Student **must** identify the essential assets required for the immersive technology prototype. To achieve this criterion, students are not required to identify all non-essential assets. This assessment criterion **could** be evidenced in an assets list.
P5	• Students **must** describe the immersive technology prototype functionality required to meet client and user requirements described in P1. The description **must** include references to triggers and layers.
P6	• Students **must** describe the testing methods and testing types they will use to test the immersive technology prototype and the elements of the immersive technology prototype they intend to test. The description of how the software solution will be tested **could** include the content in Topic Area 4.1.

M1	• Students **must** explain how the technical requirements will meet the client and user requirements detailed in P1. The explanation **must** reference hardware devices and software. Where students do not achieve P1, it is still possible to achieve M1. This assessment criterion **could** be evidenced in a requirements specification.
M2	• Students **must** plan how users will interact with the immersive technology prototype. Students **could** use the planning tools in Topic Area 2.2. The planning **must** contain enough detail for it to be interpreted by someone who hasn't seen it before.
M3	• Students **must** explain how the essential assets identified in P4 will be used in the immersive technology prototype. Students **must** explain which of the essential assets will be static and which will work in combination as triggers and layers. This assessment criterion **could** be evidenced in an assets list and/or via annotations on the designs for intended layout (P3).
M4	• Students **must** justify the approach to testing detailed in P6.
D1	• There is no assessment guidance for this criterion.
D2	• There is no assessment guidance for this criterion.
Task 2	• Ideally students will create the immersive technology prototype planned and designed in **Task 1**. However, if students deviate from the plan(s) and/or design(s) they should not be penalised when assessing **Task 2**. • To confirm assessment decisions made for this task, the OCR assessor will need to be able to see the final immersive technology prototype. Therefore, students **must**, provide either: o The final immersive technology prototype in a format which allows it to be viewed/used without the need to install any specialist software. o Video/screen recordings of the final immersive technology prototype being demonstrated. • Some software will output the immersive technology prototype, online, with a QR code to access it. In this case students must produce a video demonstrating the immersive technology prototype as a QR code to an online version is not suitable evidence.
P7	• Students must source assets for use in the immersive technology prototype. The assets sourced must be appropriate for the requirements detailed in the scenario. The final immersive technology prototype will be sufficient evidence for this assessment criterion.
P8	• Students **must** add layers to the immersive technology prototype, so it functions as intended. Students **must** use the software features and techniques in Topic Area 3.2. The final immersive technology prototype will be sufficient evidence for this assessment criterion.
P9	• Students **must** add triggers to the immersive technology prototype, so it functions as intended. Students **must** use the software features and techniques in Topic Area 3.2. The final immersive technology prototype will be sufficient evidence for this assessment criterion.
P10	• Students **must** add user interaction to the immersive technology prototype, so it functions as intended. Students **must** use the software features and techniques in Topic Area 3.2. The final immersive technology prototype will be sufficient evidence for this assessment criterion.
M5	• Students **must** prepare **all** assets sourced in P7, so they are appropriate for use in the immersive technology prototype. Topic Area 3.1 has examples of techniques students could use to prepare

	assets**.** The final immersive technology prototype will be sufficient evidence for this assessment criterion.
M6	• Students **must** combine content so there is a smooth transition from one item to another using the software features and techniques in Topic Area 3.2. The final immersive technology prototype will be sufficient evidence for this assessment criterion.
D3	• Students **must** create an immersive technology prototype where the assets and actions work cohesively together to form the immersive environment. The immersive technology prototype **must** function as intended and meet the requirements of the scenario. Students **must** use the techniques in Topic Area 3.1 and software features and techniques in Topic Area 3.2. The final immersive technology prototype will be sufficient evidence for this assessment criterion.
D4	• Students **must** implement at least **two** of the supporting technologies in Topic Area 1.2.4. The final immersive technology prototype will be sufficient evidence for this assessment criterion.
P11	• Students must test the immersive technology prototype and document results. Ideally students will use the approach described and justified in Task 1. However, if students deviate from the proposed testing they should not be penalised. • Students must have evidence of the actual test results. For example, screen shots, photographs or video/screen recordings.
P12	• Students must assess the suitability of the immersive technology prototype for meeting the requirements in Topic Area 5.1.
M7	• Students must analyse the test results generated in P11 and explain any remedial action required to resolve the issues found during testing. Students are not expected to fix errors found in the immersive prototype during final testing.
D5	• Having assessed the suitability of the immersive technology prototype (P12) and analysed test results (M7), students must discuss potential improvements and further developments to the immersive technology prototype.

Synoptic assessment

Some of the knowledge, understanding and skills needed to complete this unit will draw on the learning in Unit/s F160 and F161.

This table details these synoptic links.

Unit F165: Immersive technology solution development		Unit F160: Fundamentals of application development	
Topic Area		Topic Area	
1	Principles of immersive technology	1 5	Types of software used in application design Human computer interface and interaction
2	Plan and design high-fidelity immersive technology prototypes	2 3 4 5	Software development models Planning application developments Application design scoping Human computer interface and interaction
3	Create high-fidelity immersive technology prototypes	2 5	Software development models Human computer interface and interaction
4	Test high-fidelity immersive technology prototypes	2 5	Software development models Human computer interface and interaction
5	Review and improve the effectiveness of high-fidelity immersive technology prototypes	1 2 3 4 5	Types of software used in application design Software development models Planning application developments Application design scoping Human computer interface and interaction

Unit F165: Immersive technology solution development		Unit F161: Developing application software	
Topic Area		Topic Area	
1	Principles of immersive technology	1	Application software considerations
2	Plan and design high-fidelity immersive technology prototypes	1 2 3 4 6	Application software considerations Data and flow in application software API and protocols Application software security Legal considerations
3	Create high-fidelity immersive technology prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations
4	Test high-fidelity immersive technology prototypes	1 2 3 4 5	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations
5	Review and improve the effectiveness of high-fidelity immersive technology prototypes	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations

More information about synoptic assessment in these qualifications can be found in **Section 5.2 Synoptic assessment.**

4.3.5 Unit F166: Software development

Unit Aim

Software development is the process of designing, creating, testing, and deploying software solutions. Behind every piece of software or application, there's a set of instructions that tell the device running the software what to do. Starting from a list of what the software solution needs to do, developers design how it will look and function – the processing, storage, and output requirements. They then use a programming language to turn the designs into reality. This involves making sure it works but also pretending to be an end user and coming up with ways to try and break it.

In this unit you will learn about software design principles and different programming language types. You will learn how to use Software Design Specifications (SDS) and Software Design Documentation (SDD) to design software solutions, including their interfaces and algorithms. You will learn a programming language and how to use it to code designs to produce a working solution. You will then learn about the different types of tests and how to apply them to make sure software solutions meet the criteria, are robust and usable. Finally, you will learn how to review software solutions, including how to recommend improvements and future developments.

Unit F166: Software Development	
Topic Area 1: Fundamentals of software development	
Teaching content	**Exemplification**
1.1 Software design principles	
□ Stepwise Refinement □ Abstraction • Functional • Data • Control □ Decomposition □ Modularity □ Object-Orientated Programming (OOP) □ Maintainability □ Encapsulation • Modules • Procedures • Functions • Classes • Properties and methods	To include: □ The features and characteristics of each principle □ How each principle impacts software design
1.2 Programming languages	
□ Programming language types • Procedural • Object orientation • Functional • Scripting	To include: □ The features and characteristics of each programming language type □ Differences between each programming language type □ Advantages and disadvantages of each programming language type □ When it is appropriate to use each programming language type
Topic Area 2: Design software solutions	
Teaching content	**Exemplification**
2.1 Tools and techniques to design software solutions	
2.1.1 Software Design Specifications (SDS) □ Format, layout and templates for SDSs □ Content of SDSs • Solution overview • Client requirements • Functional requirements	To include: □ The purpose and use of SDSs □ How the format, layout and structure of SDSs impact their effectiveness □ How to create SDSs for software solutions

• Non-functional requirements • Constraints	□ How each type of requirement impacts the design of software solutions □ How constraints impact the design of software solutions Examples of **constraints** may include: □ Audience □ Budget □ Client requirements □ External dependencies □ Hardware limitations □ Industry standards □ Language limitations □ Organisational policies □ Technical requirements □ Time
2.1.2 Software Design Documentation (SDD) □ SDD components • Data structure design • Data flow diagrams ○ Level 0 ○ Level 1 • Architectural design • Interface design • Algorithm designs ○ Input ○ Process ○ Storage ○ Output □ Software design tools • Data structure design • Data flow diagrams • Interface design ○ Navigation diagram ○ Wireframe ○ Visualisation diagrams • Algorithm design ○ Flowchart ○ Pseudocode	To include: □ The purpose and use of each SDD component □ How SDD is used to design software solutions □ The purpose and use of each software design tool □ The components and conventions of each software design tool □ How each software design tool is used appropriately in SDD □ How to use software design tools to design software solutions Examples of **architectural design** may include: □ External components □ Component interfaces □ Module/component interactions □ Component-level design Examples of **interface design** may include: □ Visual representation of interface □ Navigation between interface elements □ Response time □ User help □ Error messages □ Command labelling
Topic Area 3 Create software solutions	
Teaching content	**Exemplification**
3.1 Programming techniques to develop software solutions	
3.1.1 Variables and constants □ Naming conventions • Kebab case • Camel case □ Data types • Integer • Floating point • String (or equivalent) • Boolean □ Manipulation	To include: □ The difference between variables and constants □ The purpose and use of variables and constants □ How to use naming conventions and data types when declaring variables and constants □ How to use syntax to manipulate the data type of variables and constants

• Converting between data types	
3.1.2 Operators □ Arithmetical • Plus: +, minus: -, multiplication: *, divide: /, modulus: MOD, quotient: DIV, exponentiation: ^, brackets: (), □ Boolean • AND, OR, NOT □ Relational • Less than: <, less than or equal to: <=, greater than: >, greater than or equal to: >=, equal to: ==, not equal to: !=	To include: □ The purpose and use of operators □ How to use operators within routines
3.1.3 Selection □ Selection routines • If Then Else • Else If/ElseIf • End If • Case/Switch	To include: □ The purpose and use of selection routines □ How to use syntax to create selection routines
3.1.4 Iteration □ Fixed Loop □ Conditional Loop • Pre-condition • Post condition	To include: □ The purpose and use of iteration □ How to use syntax to create iterations
3.1.5 Encapsulation □ Modules □ Procedures □ Functions □ Classes • Properties and methods □ Libraries □ Parameter passing and return values • Byref and byval • Getters and setters	To include: □ The purpose and use of modules, procedures, functions, classes and libraries □ How to use of predefined routines and libraries □ How to use parameters to pass and return data between modules, procedures, functions, classes and libraries
3.1.6 File Manipulation □ Opening and closing files □ Reading from, and writing, to files □ Managing files	To include: □ The purpose and use of file manipulation □ How to use syntax to read and write data to, and from, files □ How to use syntax to create, copy, delete and backup files
3.1.7 Data Structures □ Arrays □ Linked lists □ Stacks □ Queues	To include: □ The purpose and use of each data structure □ How to use syntax to store and retrieve data to, and from, data structures
3.1.8 Other constructs and error handling □ Other constructs • Input ○ User input ○ From file • Output from module or procedure as input ○ To file ○ To user ○ To procedure or module • Searching • Sorting	To include: □ The purpose and use of each construct □ How to use syntax to input data into software solutions □ How to use syntax to output data from software solutions □ How to use syntax to search and sort data □ How to use syntax to handle errors

□ Error handling • Try and exception • Validation rules	
3.2 Technical skills to create software solutions	
□ Development environments □ Version control • Version number • Date amended • Amended by • Amends □ Source code comments • Program headers ○ Overview of purpose of code segment • Syntax comments □ Source code indentation style	To include: □ The purpose and features of development environments □ How to use tools in development environments to create software solutions □ The importance of version control □ How to use version control when creating software solutions □ The importance of adding comments and indentation to source code □ How to use comments in source code □ How to use indentation in source code Examples of **tools in development environments** may include: □ Editor □ Autocomplete □ Keyword highlighting □ Syntax checking □ Runtime environment □ Debugging tools □ Break points □ Memory inspector
Topic Area 4: Test software solutions	
Teaching content	**Exemplification**
4.1 Software solution testing	
□ Testing methods • Dry run/trace table • Iterative • Test Plan □ Testing types • Requirements testing • Component testing • Integration testing • System testing □ Elements of software solutions to test • Input • Output • Navigation • Error handling • Data storage □ Results analysis and remedial action	To include: □ The structure, content and use of testing methods □ How and why to test iteratively both during software creation and post-software creation □ The purpose of each testing type □ The features and characteristics of each testing type □ When it is appropriate to use each testing type □ How to plan testing to ensure software solutions function as intended □ How to implement testing to ensure software solutions function as intended □ How to analyse testing results and identify remedial action Does not include: □ Implementing remedial action
Topic Area 5: Review and improve software solutions	
Teaching content	**Exemplification**
5.1 Techniques to review the fitness for purpose of software solutions	
□ Suitability for meeting: • Client requirements • Functional requirements • Non-functional requirements	To include: □ How to assess strengths and weaknesses of software solutions

□ Maintainability □ Robustness	□ How to compare software solutions against requirements □ How to assess the maintainability and robustness of software solutions
5.2 Improvements to, and further developments for, software solutions	
5.2.1 Constraints and improvements □ Constraints • Programming constructs • Language chosen • Skills of the developer • Development environment □ Improvements • Code efficiency • HCI design principles • HCI accessibility principles • Data exchange • Security	To include: □ How to assess the impact of constraints on software solutions □ How to assess potential improvements to software solutions Does not include: □ Implementing improvements to software solutions
5.2.2 Further development opportunities □ Portability of software solution □ Code reusability	To include: □ How to assess potential further development opportunities for software solutions Does not include: □ Implementing further development recommendations to software solutions

Assessment criteria

Section 6.4 provides full information on how to assess the NEA units and apply the assessment criteria.

These are the assessment criteria for the tasks for this unit. The assessment criteria indicate what is required in each task. Students' work must show that all aspects of a criterion have been met in sufficient detail for it to be **successfully achieved** (see **Section 6.4.1**). If a student's work does not fully meet a criterion, you must not award that criterion.

The command words used in the assessment criteria are defined in **Appendix B**.

Pass	Merit	Distinction
P1: Describe the solution overview for the software solution.	**M1: Explain** how the functional and non-functional requirements impact the design of the software solution.	**D1: Explain** how constraints impact the design for the software solution.
P2: Describe the client requirements for the software solution.		
P3: Create data structure and interface designs for the software solution.	**M2: Explain** how the software design documentation created allows the requirements of the SDS to be realised.	**D2: Assess** the software solution design in relation to the software design principles.
P4: Create data flow diagrams and algorithm designs for the software solution.		
P5: Describe how the software solution will be tested.	**M3: Justify** the appropriateness of the testing.	

Pass	Merit	Distinction
P6: Create a user interface for the software solution.	**M4: Use** programming techniques to implement appropriate file manipulation in the software solution.	**D3: Use** programming techniques to implement appropriate encapsulation in the software solution.
P7: Create the output(s) for the software solution.	**M5: Use** programming techniques to implement appropriate data structures in the software solution.	**D4: Use** programming techniques to implement appropriate searching and/or sorting in the software solution.
P8: Use programming techniques to implement appropriate selection and iteration in the software solution.	**M6: Use** programming techniques to implement appropriate error handling in the software solution.	
P9: Use source code comments, indentation and version control to make the software solution maintainable.		
P10: Use appropriate naming conventions and data types in the software solution.		
P11: Test the software solution and document results.	**M7: Analyse** test results documenting any required remedial action.	**D5: Discuss** potential improvements and further development opportunities for the software solution.
P12: Assess the suitability of the software solution for meeting the requirements.		

Assessment guidance

This assessment guidance gives you information relating to the assessment criteria. There might not be additional assessment guidance for each assessment criterion. It is included only where it is needed.

Assessment Criteria	Assessment guidance
P1	• Students **must** extract the specific objectives of the software solution from the scenario and describe them in a Software Design Specification (SDS).
P2	• Students **must** describe the client requirements in a Software Design Specification (SDS) expanding the descriptions into specific requirements which can be used as criteria to review against in **Task 3**.
P3	• Students **must** create data structure and interface designs for the software solution. The data structure design(s) **must** show how the data will be stored. Students **must** use at least **one** of the software design tools in Topic Area 2.1.2. The designs must contain enough detail for them to be interpreted by someone who hasn't seen them before.
P4	• Students **must** create data flow diagrams to show how data will flow through the software solution. Students **must** use at least **one** of the software design tools in Topic Area 2.1.2 to create algorithm designs for the software solution. The designs must contain enough detail for them to be interpreted by someone who hasn't seen them before.
P5	• Students **must** describe the testing methods and testing types they will use to test the software solution and the elements of the software solution they intend to test. The description of how the software solution will be tested **could** include the content in Topic Area 4.1.

M1	• M1 is an extension of P1 and P2. Students **must** explain how the functional and non-functional requirements of the solution will influence the design of the software solution.
M2	• Students **must** explain how each of the designs created in P3 and P4 (data structure, interface, data flow and algorithms) relate to the requirements detailed in the SDS.
M3	• Students **must** justify the approach to testing detailed in P5.
D1	• Students **must** consider at least **three** potential constraints and explain how they would impact the design of the software solution.
D2	• Students **must** assess the extent to which the software design principles in Topic Area 1.1 have been applied to the software solution design. The criterion is achieved if students consider at least **three** of the software design principles.
Task 2	• Ideally, students will create the software solution designed in **Task 1**. However, if students deviate from the design(s) they should not be penalised when assessing **Task 2.** • To confirm assessment decisions made for some of the criteria for this task, the OCR assessor will need to be able to see the final software solution. Therefore, students **must**, provide either: o The final software solution in a format which allows it to be viewed/used without the need to install any specialist software. o Video/screen recordings of the final software solution being demonstrated.
P6	• Students **must** create a user interface for the software solution. The final software solution will be sufficient evidence for this assessment criterion.
P7	• Students **must** create the output(s) for the software solution. The final software solution will be sufficient evidence for this assessment criterion.
P8	• Students **must** use the programming techniques in Topic Area 3.1 (as required) to add selection and iteration to the software solution, so it functions as intended. The source code from the final software solution will be sufficient evidence for this assessment criterion.
P9	• Students **must** use the technical skills in Topic Area 3.2 (as required) to ensure the code is maintainable. For the code to be maintainable, someone who hasn't seen it before **must** be able to be interpret it. This assessment criterion **could** be evidenced by the source code from the final software solution and screen shots or photographs showing the use of version control.
P10	• Students **must** use a consistent and understandable naming convention for variables, constants, files, data structures and encapsulation. For the naming conventions to be understandable, someone who hasn't seen the code before **must** be able to be interpret it. The source code from the final software solution will be sufficient evidence for this assessment criterion.
M4	• Students **must** use the programming techniques in Topic Area 3.1 (as required) to add file manipulation to the software solution, so it functions as intended. The source code from the final software solution will be sufficient evidence for this assessment criterion.
M5	• Students **must** use the programming techniques in Topic Area 3.1 (as required) to create the data structures for the software solution, so it functions as intended. The source code from the final software solution will be sufficient evidence for this assessment criterion.
M6	• Students **must** use the programming techniques in Topic Area 3.1 (as required) to add error handling errors in relation to user input and

	to prevent software solutions from unexpected and unintended closure. The source code from the final software solution will be sufficient evidence for this assessment criterion.
D3	• Students must use the programming techniques in Topic Area 3.1 (as required) to add encapsulation that improves the efficiency of the software solution. The source code from the final software solution will be sufficient evidence for this assessment criterion.
D4	• Students must use the programming techniques in Topic Area 3.1 (as required) to add searching and/or sorting to the software solution, so it functions as intended. The source code from the final software solution will be sufficient evidence for this assessment criterion.
P11	• Students must test the software solution and document results. Ideally students will use the approach described and justified in Task 1. However, if students deviate from the proposed testing they should not be penalised. • Students must have evidence of the actual test results for example screen shots, photographs or video/screen recordings.
P12	• Students must assess the suitability of the software solution for meeting the requirements in Topic Area 5.1.
M7	• Students must analyse the test results generated in P11 and explain any remedial action required to resolve the issues found during testing. Students are not expected to fix errors found in the software solution during final testing.
D5	• Having assessed the suitability of the software solution (P12) and analysed test results (M7), students must discuss potential improvements and further developments to the software solution.

Synoptic assessment

Some of the knowledge, understanding and skills needed to complete this unit will draw on the learning in Unit/s F160 and F161.

This table details these synoptic links.

Unit F166: Software development		Unit F160: Fundamentals of application development	
Topic Area		Topic Area	
2	Design software solutions	2 3 4 5	Software development models Planning application developments Application design scoping Human computer interface and interaction
3	Create software solutions	2 5	Software development models Human computer interface and interaction
4	Test software solutions	2 5	Software development models Human computer interface and interaction
5	Review and improve software solutions	1 2 3 4 5	Types of software used in application design Software development models Planning application developments Application design scoping Human computer interface and interaction

Unit F166: Software development		Unit F161: Developing application software	
Topic Area		Topic Area	
2	Design software solutions	1 2 3 4 6	Application software considerations Data and flow in application software API and protocols Application software security Legal considerations
3	Create software solutions	1 2 3 4 5	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations
4	Test software solutions	1 2 3 4 5	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations
5	Review and improve software solutions	1 2 3 4 5 6	Application software considerations Data and flow in application software API and protocols Application software security Operational considerations Legal considerations

More information about synoptic assessment in these qualifications can be found in **Section 5.2 Synoptic assessment.**

5 Assessment and grading

5.1 Overview of the assessment

Entry code	H029
Qualification title	OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate)
GLH	150*
Reference	610/3974/3
Total Units	Has two units: • Mandatory units F160 and F162

Entry code	H129
Qualification title	OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate)
GLH	360*
Reference	610/3975/5
Total Units	Has five units: • Mandatory units F160, F161, F162 • And two other units from F163, F164, F165, F166.

*the GLH includes assessment time for each unit

Unit F160: Fundamentals of application development
75 GLH 1 hour 15 minute written exam 60 marks (60 UMS) OCR-set and marked Calculators are not required in this exam The exam will always **have:** • A short scenario which will develop through the paper • Forced choice/controlled response questions • Short answer, closed response questions • Extended constructed response questions with points-based marks schemes • Extended constructed response questions with levels of response marks schemes • One six mark and one nine mark extended constructed response question with a levels of response marks scheme

Unit F161: Developing application software

70 GLH

1 hour 15 minute written exam

60 marks (60 UMS)

OCR-set and marked

Calculators are not required in this exam

The exam will always **have:**

- A short scenario which will develop through the paper
- Forced choice/controlled response questions
- Short answer, closed response questions
- Extended constructed response questions with points-based marks schemes
- Extended constructed response questions with levels of response marks schemes
- One six mark and one nine mark extended constructed response question with a levels of response marks scheme

Unit F162: Designing and communicating UX/UI solutions

75 GLH

OCR-set assignment

Centre-assessed and OCR-moderated

This set assignment has four practical tasks

It should take about 15 GLH to complete

Unit F163: Game development

70 GLH

OCR-set assignment

Centre-assessed and OCR-moderated

This set assignment has three practical tasks

It should take about 15 GLH to complete

Unit F164: Website development

70 GLH

OCR-set assignment

Centre-assessed and OCR-moderated

This set assignment has three practical tasks

It should take about 15 GLH to complete

Unit F165: Immersive technology solution development
70 GLH OCR-set assignment Centre-assessed and OCR-moderated This set assignment has three practical tasks It should take about 15 GLH to complete

Unit F166: Software development
70 GLH OCR-set assignment Centre-assessed and OCR-moderated This set assignment has three practical tasks It should take about 15 GLH to complete

OCR-set assignments for NEA units are on our secure website, **Teach Cambridge**. Each NEA assignment is live for two years. The intended cohort is shown on the front cover. It is important you use the correct NEA set assignment for each cohort, as starting a new cohort of Year 12 students on an NEA set assignment that has already been live for one year will mean that these students will only have one year to work on the assignment.

5.2 Synoptic assessment

Synoptic assessment is a built-in feature of these qualifications. It means that students need to use an appropriate selection of their knowledge, understanding and skills developed across each qualification in an integrated way and apply them to a key task or tasks.

This helps students to build a holistic understanding of the subject and the connections between different elements of learning, so they can go on to apply what they learn from these qualifications to new and different situations and contexts.

The externally assessed units allow students to gain underpinning knowledge and understanding relevant to application development. The NEA units draw on and strengthen this learning by assessing it in a practical way.

It is important to be aware of the synoptic links between the units so that teaching, learning and assessment can be planned accordingly. Then students can apply their learning in ways which show they are able to make connections across the qualification. **Section 4.3** shows the synoptic links for each unit.

5.3 Transferable skills

These qualifications give students the opportunity to gain broad, transferable skills and experiences that they can apply in future study, employment and life.

Higher Education Institutions (HEIs) have told us that developing some of these skills helps students to transition into higher education.

These skills include:

- Communication
- Creativity
- Critical thinking
- Independent learning
- Presentation skills
- Problem solving
- Reflection
- Research skills
- Risk taking, resilience
- Self-directed study
- Time management.

5.4 Grading and awarding grades

Externally assessed units

We mark all the externally assessed units.

Each external assessment is marked according to a mark scheme, and the mark achieved will determine the unit grade awarded (Pass, Merit or Distinction). We determine grade boundaries for each of the external assessments in each assessment series.

If a student doesn't achieve the mark required for a Pass grade, we issue an unclassified result for that unit. The marks achieved in the external assessment will contribute towards the student's overall qualification grade, even if a Pass is not achieved in the unit assessment.

NEA units

NEA units are assessed by the teacher and externally moderated by us.

Each unit has specified Pass, Merit and Distinction assessment criteria. The assessment criteria for each unit are provided with the unit content in **Section 4.3** of this specification. Teachers must judge whether students have met the criteria or not.

A unit grade can be awarded at Pass, Merit or Distinction. The number of assessment criteria needed to achieve each grade has been built into each assignment. These are referred to as design thresholds. The table below shows the design thresholds for each grade outcome for the NEA assessments in these qualifications. The unit grade awarded is based on the **total** number of achieved criteria for the unit. The total number of achieved criteria for each unit can come from achievement of any of the criteria (Pass, Merit or Distinction). This is **not** a 'hurdles-based' approach, so students do **not** have to achieve **all** criteria for a specific grade to achieve that grade (e.g. all Pass criteria to achieve a Pass).

To make sure we can keep outcomes fair and comparable over time, we will review the performance of the qualifications through their lifetime. The review process might lead to changes in these design thresholds if any unexpected outcomes or significant changes are identified.

Unit size (GLH)	70	75
Total number of criteria	24	24
Number of pass criteria	12	12
Number of merit criteria	7	7
Number of distinction criteria	5	5
Total number of criteria needed for a unit pass	10	10
Total number of criteria needed for a unit merit	15	15
Total number of criteria needed for a unit distinction	20	20

If a student doesn't achieve enough criteria to achieve a unit Pass, we will issue an unclassified result for that unit. The number of criteria achieved will be converted into a mark on the Uniform Mark Scale (UMS) and will contribute towards the student's overall qualification grade, even if a Pass is not achieved in the unit assessment. More information about this is in Section below (**Calculating the qualification grades**).

Qualifications

The overall qualification grades are:

- Distinction* (D*)
- Distinction (D)
- Merit (M)
- Pass (P)
- Unclassified (U)

Calculating the qualification grades

When we work out students' overall grades, we need to be able to compare performance on the same unit in different assessments over time and between different units. We use a Uniform Mark Scale (UMS) to do this.

A student's uniform mark for each externally assessed unit is calculated from the student's raw mark on that unit. A student's uniform mark for each NEA unit is calculated from the number of criteria the student achieves for that unit. The raw mark or number of criteria achieved are converted to the equivalent mark on the uniform mark scale. Marks between grade boundaries are converted on a pro rata basis.

When unit results are issued, the student's unit grade and uniform mark are given. The uniform mark is shown out of the maximum uniform mark for the unit (for example, 48/60).

The student's uniform marks for each unit will be aggregated to give a total uniform mark for the qualification. The student's overall grade will be determined by the total uniform mark.

The tables below show:

- the maximum raw marks or number of criteria, and uniform marks for each unit in the qualifications
- the uniform mark boundaries for each of the assessments in each qualification
- the minimum total mark for each overall grade in the qualifications.

Certificate Qualification:

Unit	Maximum raw mark/ number of criteria	Maximum uniform mark (UMS)	Distinction* (UMS)	Distinction (UMS)	Merit (UMS)	Pass (UMS)
F160	60	60	-	48	36	24
F162	24	60	-	48	36	24
Qualification Totals	84	120	108	96	72	48

Extended Certificate Qualification:

Unit	Maximum raw mark/ number of criteria	Maximum uniform mark (UMS)	Distinction* (UMS)	Distinction (UMS)	Merit (UMS)	Pass (UMS)
F160	60	60	-	48	36	24
F161	60	60	-	48	36	24
F162	24	60	-	48	36	24
F163	24	60	-	48	36	24
F164	24	60	-	48	36	24
F165	24	60	-	48	36	24
F166	24	60	-	48	36	24
Qualification Totals	192	300	270	240	180	120

You can find a marks calculator on the qualification page of the OCR website to help you convert raw marks/number of achieved criteria into uniform marks.

5.5 Performance descriptors

Performance descriptors indicate likely levels of attainment by representative students performing at the Pass, Merit and Distinction grade boundaries at Level 3.

The descriptors must be interpreted in relation to the content in the units and the qualification as a whole. They are not designed to define that content. The grade achieved will depend on how far the student has met the assessment criteria overall. Shortcomings in some parts of the assessment might be balanced by better performance in others.

Level 3 Pass

At Pass, students show adequate knowledge and understanding of the basic elements of much of the content being assessed. They can develop and apply their knowledge and understanding to some basic and familiar contexts, situations and problems.

Responses to higher order tasks involving detailed discussion, evaluation and analysis are often limited.

Many of the most fundamental skills and processes relevant to the subject are executed effectively but lack refinement, producing functional outcomes. Demonstration and application of more advanced skills and processes might be attempted but not always executed successfully.

Level 3 Merit

At Merit, students show good knowledge and understanding of many elements of the content being assessed. They can sometimes develop and apply their understanding to different contexts, situations and problems, including some which are more complex or less familiar.

Responses to higher order tasks involving detailed discussion, evaluation and analysis are likely to be mixed, with some good examples at times and others which are less accomplished.

Skills and processes relevant to the subject, including more advanced ones, are developed in terms of range and quality. They generally lead to outcomes which are of good quality, as well as being functional.

Level 3 Distinction

At Distinction, students show thorough knowledge and understanding of most elements of the content being assessed. They can consistently develop and apply their understanding to different contexts, situations and problems, including those which are more complex or less familiar.

Responses to higher order tasks involving detailed discussion, evaluation and analysis are successful in most cases.

Most skills and processes relevant to the subject, including more advanced ones, are well developed and consistently executed, leading to high quality outcomes.

6 Non examined assessment (NEA) units

This section gives guidance on completing the NEA units. In the NEA units, students build a portfolio of evidence to meet the assessment criteria for the unit.

Assessment for these qualifications **must** adhere to JCQ's **Instructions for Conducting Coursework**. Do **not** use JCQ's Instructions for Conducting Non-examination Assessments – these are only relevant to GCE and GCSE specifications.

The NEA units are centre-assessed and externally moderated by us.

You **must** read and understand all the rules and guidance in this section **before** your students start the set assignments.

If you have any questions, please contact us for help and support.

6.1 Preparing for NEA unit delivery and assessment

6.1.1 Centre and teacher/assessor responsibilities

We assume the teacher is the assessor for the NEA units.

Before you apply to us for approval to offer these qualifications you must be confident your centre can fulfil all the responsibilities described below. Once you're approved, you can offer any of our general qualifications, Cambridge Nationals or Cambridge Advanced Nationals **without** having to seek approval for individual qualifications.

Here's a summary of the responsibilities that your centre and teachers must be able to fulfil. It is the responsibility of the head of centre[1] to make sure our requirements are met. The head of centre must ensure that:

- there are enough trained or qualified people to teach and assess the expected number of students you have in your cohorts.
- teaching staff have the relevant level of subject knowledge and skills to deliver and assess these qualifications.
- teaching staff will fully cover the knowledge, understanding and skills requirements in teaching and learning activities.
- allowed combinations of units are considered at the start of the course to be confident that all students can access a valid route through the qualifications.
- all necessary resources are available for teaching staff and students during teaching and assessment activities. This gives students every opportunity to meet the requirements of the qualification and reach the highest grade possible.
- there is a system of internal standardisation in place so that all assessment decisions for centre-assessed assignments are consistent, fair, valid and reliable (see **Section 6.4.3**).
- there is enough time for effective teaching and learning, assessment and internal standardisation.
- processes are in place to make sure that students' work is individual and confirmed as authentic (see **Section 6.2.1**).

[1] This is the most senior officer in the organisation, directly responsible for the delivery of OCR qualifications, For example, the headteacher or principal of a school/college. The head of centre accepts full responsibility for the correct administration and conduct of OCR exams.

- OCR-set assignments are used for students' summative assessments.
- OCR-set assignments are **not** used for practice. Sample assessment material for each of the NEA units is available on the OCR website. This sample assessment material can be used for practice purposes.
- students understand what they need to do to achieve the criteria.
- students understand what it means when we say work must be authentic and individual and they (and you) follow our requirements to make sure their work is their own.
- students know they must not reference another individual's personal details in any evidence produced for summative assessment, in accordance with the Data Protection Act 2018 and the UK General Data Protection Regulations (UK GDPR). It is the student's responsibility to make sure evidence that includes another individual's personal details is anonymised.
- outcomes submitted to us are correct and are accurately recorded.
- assessment of set assignments adheres to the JCQ **Instructions for Conducting Coursework** and JCQ **AI Use in Assessments: Protecting the Integrity of Qualifications**.
- a declaration is made at the point you're submitting any work to us for assessment that confirms:
    - all assessment is conducted according to the specified regulations identified in the **Administration** area of our website,
    - students' work is authentic.
    - marks have been transcribed accurately.
- centre records and students' work are kept according to these requirements:
    - students' work **must** be kept until **after** the unit has been awarded and any review of results or appeals processed. We cannot consider any review if the work has not been kept.
    - internal standardisation and assessment records must be kept securely for a minimum of three years after the date we've issued a certificate for a qualification.
- all cases of suspected malpractice involving teachers or students are reported (see **Section 6.3.1**).

6.2 Requirements and guidance for delivering and marking the OCR-set assignments

The assignments are:

- set by us.
- taken under supervised conditions (unless we specify otherwise in the assessment guidance)
- assessed by the teacher.
- moderated by us.

You can find the set assignments on our secure website, **Teach Cambridge**.

The set assignments give an approximate time that it will take to complete all the tasks. These timings are for guidance only, but should be used by you, the teacher, to give students an indication of how long to spend on each task. You can decide how the time should be allocated between each task or part task. Students can complete the tasks and produce the evidence across several sessions. Students' evidence (either hard copy or digital) must be kept securely by the teacher and access to assessment responses must be controlled. Students aren't permitted to access their work in between the assessment sessions.

We will publish a new set assignment each year and they will be live for 2 years. Each new set assignment will be released on 1 June. You must check our secure website, **Teach Cambridge**, and use a set assignment that is live for assessment. The live assessment dates will be shown on the front cover alongside the intended cohort. You should use the set assignment released in the same calendar year as the new cohort starts to ensure they have two years for that assignment. Students are allowed one resubmission of work based on the same live assignment.

You must have made unit entries before submitting NEA work for moderation.

Appendix A of this specification gives guidance for creating electronic evidence for the NEA units. Read Appendix A in conjunction with the unit content and assessment criteria grids to help you plan the delivery of each unit.

The rest of this section is about how to manage the delivery and marking of the set assignments so that assessment is valid and reliable. Please note that failing to meet these requirements might be considered as malpractice.

Here is a summary of what you need to do.

You **must**:

- have covered the knowledge, understanding and skills with your students and be sure they are ready for assessment **before** you start the summative assessment. This may include students practising applying their learning and receiving feedback from teachers in preparing to take the assessment.
- use an OCR-set assignment for summative assessment of the students.
- give students the **Student Guide** before they start the assessment.
- familiarise yourself with the assessment guidance relating to the tasks. The assessment guidance for each unit is in **Section 4** after the assessment criteria grids and with the student tasks in the assignments.
- make sure students are clear about the tasks they must complete and the assessment criteria they are attempting to meet.

- give students a reasonable amount of time to complete the assignments and be fair and consistent to all students. The estimated time we think each assignment should take is stated in the OCR-set assignments. In that time students can work on the tasks under the specified conditions until the date that you collect the work for centre assessment.
- tell the students the resources they can use in the assignment before they start the assessment tasks.
- only give students OCR-provided templates. If they choose to use a different template from a book, a website or course notes (for example, to create a plan) they **must** make sure the source is referenced.
- monitor students' progress to make sure work is capable of being assessed against the assessment criteria, on track for being completed in good time and is the student's own work:
    - NEA work must be completed in the centre under teacher supervision in normal curriculum time:
        - work must be completed with enough supervision to make sure that it can be authenticated as the student's own work. You must be familiar with the requirements of the JCQ document **AI Use in Assessments: Protecting the Integrity of Qualifications** before assessment starts.
        - there may be exceptions to the requirement for supervised conditions if there is work to complete to support the assignment tasks (e.g. research). The assignment and assessment guidance will specify if there are exceptions.

          Where students are allowed to complete work outside of supervised conditions (e.g. research that may be allowed between supervised sessions) you **must** make sure that they only bring notes relating to the work they are allowed to complete unsupervised into the supervised sessions (e.g. notes relating to the research they have done) and to make sure any work they have done is independent. They **must not** use unsupervised time as an opportunity to:

          - Create drafts of work for their tasks.
          - Gather information to use in other aspects of their tasks.
        - if you provide any material to prepare students for the set assignment, you must adhere to the rules on using referencing and on acceptable levels of guidance to students. This is in section **6.2.3 and 6.3**.
        - students must produce their work independently (see sections **6.2.1 and 6.3**).
        - you must make sure students know to keep their work and passwords secure and know that they must not share completed work with other students, use any aspect of another student's work or share their passwords.
- complete the **Teacher Observation Record** that is with the assignments for tasks that state it is needed. You **must** follow the guidance given when completing it.
- use the assessment criteria to assess students' work.
- before submitting a final outcome to us, you can allow students to repeat any part of the assignment and rework their original evidence. But any feedback you give to students on the original (assessed) evidence, must:
    - only be generic.
    - be recorded.
    - be available to the OCR assessor.

(See **Section 6.3** on **Feedback** and **Section 6.4.4** on **resubmitting work**).

You **must not**:

- change any part of the OCR-set assignments (scenarios or tasks).
- accept multiple reattempts of work where small changes have been made in response to feedback.
- allow teachers or students to add, amend or remove any work **after** submission for moderation by OCR. This will constitute malpractice.
- give detailed advice and suggestions to individuals or the whole class on how work may be improved to meet the assessment criteria. This includes giving access to student work as an exemplar.
- allow students access to their assignment work between teacher supervised sessions. (There may be exceptions where students are allowed to complete work independently (e.g. research). Any exceptions will be stated in the assignments.)
- practise the live OCR-set assignment tasks with the students.

6.2.1 Ways to authenticate work

You must use enough supervision and complete enough checks to be confident that the work you mark is the student's own and was produced independently.

Where possible, you should discuss work in progress with students. This will make sure that work is being completed in a planned and timely way and will give you opportunities to check the authenticity of the work.

You **must**:

- have read and understood the JCQ document **AI Use in Assessments: Protecting the Integrity of Qualifications**.
- make sure students and other teachers understand what constitutes plagiarism.
- not accept plagiarised work as evidence.
- use supervision and questioning as appropriate to confirm authenticity.
- make sure students and teachers fill in declaration statements.

6.2.2 Group work

Group work is not allowed for the NEA assignments in these qualifications.

6.2.3 Plagiarism

Students must use their own words when they produce final written pieces of work to show they have genuinely applied their knowledge and understanding. When students use their own words, ideas and opinions, it reduces the possibility of their work being identified as plagiarised. Plagiarism is:

- the submission of someone else's work as your own
- failure to acknowledge a source correctly, including any use of Artificial Intelligence (AI).

You might find the following JCQ documents helpful:

- **Plagiarism in Assessments**
- **AI Use in Assessments: Protecting the Integrity of Qualifications**

Due to increasing advancements in AI technology, we strongly recommend that you are familiar with the likely outputs from AI tools. This could include using AI tools to produce responses to some of the assignment tasks, so that you can identify typical formats and wording that these may produce. This may help you identify any cases of potential plagiarism from students using AI tools to generate written responses.

Plagiarism makes up a large percentage of cases of suspected malpractice reported to us by our assessors. You must **not** accept plagiarised work as evidence.

Plagiarism often happens innocently when students do not know that they must reference or acknowledge their sources or aren't sure how to do this. It's important to make sure your students understand:

- the meaning of plagiarism and what penalties may be applied.
- that they can refer to research, quotations or evidence produced by somebody else, but they must list and reference their sources and clearly mark quotations.
- quoting someone else's work, even when it's properly sourced and referenced, doesn't evidence understanding. The student must 'do' something with that information to show they understand it. For example, if a student has to analyse data from an experiment, quoting data doesn't show that they understand what it means. The student must interpret the data and, by relating it to their assignment, say what they think it means. The work must clearly show how the student is using the material they have referenced to inform their thoughts, ideas or conclusions.

We have **The OCR Guide to Referencing** on our website. We have also produced a **poster** about referencing and plagiarism which may be useful to share with your students.

Teach your students how to reference and explain why it's important to do it. At Key Stage 5 they must:

- use quote marks to show the beginning and end of the copied work.
- list the html address for website text and the date they downloaded information from the website.
- for other publications, list:
    - the name of the author.
    - the name of the resource/book/printed article.
    - the year in which it was published.
    - the page number.

Teach your students to:

- always reference material copied from the internet or other sources. This also applies to infographics (graphical information providing data or knowledge).
- always identify information they have copied from teaching handouts and presentations for the unit, using quote marks and stating the text is from class handouts.

Identifying copied/plagiarised work

Inconsistencies throughout a student's work are often indicators of plagiarism. For example:

- different tones of voice, sentence structure and formality across pieces of work.
- use of American expressions, spellings and contexts (such as American laws and guidelines).
- dated expressions and references to past events as being current.
- sections of text in a document where the font or format is inconsistent with other sections.

What to do if you think a student has plagiarised

If you identify plagiarised work during assessment or internal standardisation, you must:

- consider the plagiarism when judging the number of assessment criteria achieved.
    - if the work is part of the moderation sample, it must be included with the other work provided to the OCR assessor. You must add a note on the Unit Recording Sheet to state that there is plagiarism in the work and the number of criteria achieved has been adjusted accordingly.
- report the student(s) for plagiarism in line with the JCQ document **Suspected Malpractice Policies and Procedures**
    - fill in the **JCQ form M1**.

In line with JCQ's policies and procedures on suspected malpractice, the penalties applied for plagiarism will usually result in the work not being allowed or the mark being significantly reduced.

6.3 Feedback

Feedback to students on work in progress towards summative assessment

You can discuss work in progress towards summative assessment with students to make sure it's being done in a planned and timely way. It also provides an opportunity to check the authenticity of the work. You must intervene if there's a health and safety risk (and reflect this in your assessment if the student's ability to operate safely and independently if that is part of the criteria).

Generic guidance to the whole class is also allowed. This could include reminding students to check they have provided evidence to cover all key aspects of the task. Individual students can be prompted to double check for gaps in evidence providing that specific gaps are not pointed out to them.

You can give general feedback and support if one or more students are struggling to get started on an aspect of the assignment or following a break between sessions working on the assignment. For example, if a student is seeking more guidance that suggests they are not able to apply knowledge, skills and understanding to complete their evidence, you can remind them that they had a lesson which covered the topic. The student would then need to review their own notes to find this information and apply it as needed.

If a student needs additional help to get started on an initial task that is critical to accessing the rest of the assessment, you can provide this help if you feel it is necessary, but you must not award the student with any assessment criteria directly associated with the part(s) of the task for which they received help.

With the exception of the specific feedback allowed to help students start a critical task, mentioned above, feedback must not provide specific advice and guidance that would be construed as coaching. This would compromise the student's ability to independently perform the task(s) they

are doing and constitutes malpractice. Our assessors use a number of measures to assure themselves the work is the student's own.

Once work has been assessed, you must give feedback to students on the work they submitted for assessment.

Feedback **must**:

- be supportive, encouraging and positive.
- tell the student what has been noticed, not what the teacher thinks (for example, if you have observed the student completing a task, you can describe what happened, what was produced and what was demonstrated).

Feedback **can**:

- identify what task and part of the task could be improved, but not say how to improve it. You could show the student work from a **different** unit that demonstrates higher achievement, but you must not detail to the student how they could achieve that in their work. If you are using another student's work from a different unit as an example, you must anonymise this work and make sure that the potential to plagiarise from this work is minimised. You could remind students that they had a lesson on a specific topic and that they could review their notes, but you must not tell them how they could apply the teaching to improve their work.
- comment on what has been achieved, for example 'the evidence meets the P2 and M2 criteria'.
- identify that the student hasn't met a command word or assessment criteria requirement. For example, 'This is a description, not an evaluation'.
- use text from the specification, assignment or assessment criteria in general guidance to clarify what is needed in the work. For example, 'M3 requires you to describe how at least **four** of Schneiderman's 8 Golden Rules of interface design have informed the design of your UX/UI solution.'

Feedback **must not**:

- point out specific gaps. For example, you must not prompt the student to include specific detail in their work, such as 'The description in P8 only describes how the UX/UI solution will meet the needs of the patients. You need to also describe how the needs of dentists and the hygienist have been met.'
- be so detailed that it leads students to the answer. For example, you must not give:
    - model answers.
    - step-by-step guidance on what to do to complete or improve work.
    - headings or templates that include examples which give all or part of what students have to write about or produce.
- talk the student through how to achieve or complete the task.
- give detail on where to find information/evidence.

In other words, feedback must help the student to take the initiative in making changes. It must not direct or tell the student what to do to complete or improve their work in a way that means they do not need to think how to apply their learning. Students need to recall or apply their learning. You must not do the work for them.

Neither you nor the student can add, amend or remove any work after the final mark has been submitted for moderation.

Sections 6.4.4 and **6.4.6** give more guidance for students who wish to reattempt or resubmit their work following feedback.

What over-direction might look like

When we see anything that suggests the teacher has led students to the answer, we become concerned because it suggests students have not worked independently to produce their assignment work. The following are examples of what might indicate over-direction by the teacher:

- prompts that instruct students to include specific detail in their work, such as, 'You need to include the aims of the activity. Who is it aimed at? What is the purpose of the activity? How will it benefit the specific group/individual?
- headings or templates that include examples which give all or part of what students have to write about or produce, such as sources of support.

OCR Assessors will report suspected malpractice when they cannot see differences in content between students' work in the sample they are moderating. An exception is when students have only used and referenced technical facts and definitions. If the OCR assessor is in any doubt, they will report suspected malpractice. The decision to investigate or not is made by us, not the assessor.

6.3.1 Reporting suspected malpractice

It is the responsibility of the head of centre to report all cases of suspected malpractice involving teachers or students.

A JCQ Report of Suspected Malpractice form (JCQ/M1 for student suspected malpractice or JCQ/M2 for staff suspected malpractice) is available to download from the **JCQ website**. The form must be completed as soon as possible and emailed to us at **malpractice@ocr.org.uk**.

When we ask centres to gather evidence to assist in any malpractice investigation, heads of centres must act promptly and report the outcomes to us.

The JCQ document **Suspected Malpractice Policies and Procedures** has more information about reporting and investigating suspected malpractice, and the possible sanctions and penalties which could be imposed. You can also find out more on our **website.**

6.3.2 Student and centre declarations

Both students and teachers must declare that the work is the student's own:

- **each student** must sign a declaration before submitting their work to their teacher. A **candidate authentication statement** can be used and is available to download from our **website**. You must keep these statements in the centre until all enquiries about results, malpractice and appeal issues have been resolved. You **must** record a mark of zero if a student cannot confirm the authenticity of their work.
- **teachers** must declare the work submitted for centre assessment is the students' own work by completing a **centre authentication form (CCS160)** for each cohort of students for each unit. You must keep centre authentication forms in the centre until all post-results issues have been resolved.

6.3.3 Generating evidence

The set assignments will tell the students what they need to do to meet the assessment criteria for the NEA units. It is your responsibility to make sure that the methods of generating evidence for the assignments are:

- valid
- safe and manageable
- suitable to the needs of the student.

Valid

The evidence presented must be valid. For example, it would not be appropriate to present an organisation's equal opportunities policy as evidence towards a student's understanding of how the equal opportunities policy operates in an organisation. It would be more appropriate for the student to incorporate the policy in a report describing the different approaches to equal opportunities.

Safe and manageable

You must make sure that methods of generating evidence are safe and manageable and do not put unnecessary demands on the student.

Suitable to the needs of the student

We are committed to ensuring that achievement of these qualifications is free from unnecessary barriers.

Observation and questioning

The primary evidence for assessment is the work submitted by the student, however the following assessment methods might be suitable for teachers/assessors to use for some aspects of these qualifications, where identified:

- **observation** of a student doing something
- **questioning** of the student or witness.

Observation

The teacher/assessor and student should plan observations together, but it is the teacher's/assessor's responsibility to record the observation properly (for example observing a student undertaking a practical task). More information is in the Teacher Observation Records section.

Questioning

Questioning the student is normally an ongoing part of the formative assessment process and may, in some circumstances, provide evidence to support achievement of the criteria.

Questioning is often used to:

- test a student's understanding of work which has been completed outside of the classroom
- check if a student understands the work they have completed
- collect information on the type and purpose of the processes a student has gone through.

If questioning is used as evidence towards achievement of specific topic areas, it is important that teachers/assessors record enough information about what they asked and how the student replied, to allow the assessment decision to be moderated.

6.3.4 **Teacher Observation Records**

You **must** complete the Teacher Observation Record form in the OCR-set assignment for:

Unit F162 for each student as evidence of effective communication while delivering the UX/UI showcase (Task 3, Topic Area 4). The Teacher Observation Record form must provide evidence that students have clearly demonstrated at least **three** of the techniques in Topic Area 4.2 (clarity, coherence, completeness, conciseness, correctness, courteousness). For other criterions in this task students **must** provide suitable evidence in the form of a slide deck, screenshots, photographs, screen recordings, presentation notes, a script, supporting visual stimuli, for example. There is no requirement to submit audio or visual recordings of students delivering UX/UI showcases.

Teacher observation **cannot** be used as evidence of achievement for a whole unit. Most evidence **must** be produced directly by the student. Teacher observation **must only** be used where specified as an evidence requirement.

Teacher Observation Records must be suitably detailed for each student, to help assessors to determine if the assessment criteria have been met. You must follow the guidance provided in the 'guidance notes' section of the form so that the evidence captured and submitted is appropriate. Both you and the student must sign and date the form to show that you both agree its contents.

Where the guidance has not been followed, the reliability of the form as evidence may be called into question. If doubt about the validity of the Teacher Observation Record form exists, it cannot be used as assessment evidence and marks based on it cannot be awarded. OCR assessors will be instructed to adjust centre marks accordingly.

6.3.5 **Presentation of the final piece of work**

Students must submit their evidence in the format specified in the tasks where specific formats are given. Written work can be digital (e.g. word processed) or hand-written and tables and graphs (if relevant) can be produced using appropriate ICT.

Any sourced material must be suitably acknowledged. Quotations must be clearly marked and a reference provided.

A completed Unit Recording Sheet (URS) must be attached to work submitted for moderation.

The URS can be downloaded from the qualification webpage. Centres **must** show on the URS where specific evidence can be found. The URS tells you how to do this.

Work submitted digitally for moderation should be on electronic media (for example, on our portal, CD or USB Drive). Work **must** be in a suitable file format and structure. **Appendix A** gives more guidance about submitting work in digital format.

6.4 Assessing NEA units

All NEA units are assessed by teachers and externally moderated by OCR assessors. Assessment of the set assignments must adhere to JCQ's **Instructions for Conducting Coursework**.

The centre is responsible for appointing someone to act as the internal assessor. This would usually be the teacher who has delivered the programme but could be another person from the centre. The assessment criteria must be used to assess the student's work. These specify the levels of skills, knowledge and understanding that the student needs to demonstrate.

6.4.1 **Applying the assessment criteria**

When students have completed the assignment, they must submit their work to you to be assessed.

You must assess the tasks using the assessment criteria and any additional assessment guidance provided. Each criterion states what the student needs to do to achieve that criterion (e.g. **Describe** the client and user requirements for the UX/UI solution). The command word and assessment guidance provide additional detail about breadth and depth where it is needed.

You must judge whether each assessment criterion has been **successfully achieved** based on the evidence that a student has produced. For the criterion to be achieved, the evidence must show that all aspects have been met in sufficient detail.

When making a judgement about whether a criterion has been **successfully achieved**, you must consider:

- the requirements of the NEA task
- the criterion wording, including the command word used and its definition
- any assessment guidance for the criterion
- the unit content that is being assessed.

You must annotate the work to show where evidence meets each criterion (see **Section 6.4.2**). You can then award the criterion on the Unit Recording Sheet (URS). Assessment should be positive, rewarding achievement rather than penalising failure or omissions.

The number of criteria needed for each unit grade (Pass, Merit or Distinction) is provided in **Section 5**

You must complete a Unit Recording Sheet (URS) for each unit a student completes. On the URS you must identify:

- whether the student has met each criterion or not (by adding a tick (✓) or X in the column titled **Assessment criteria achieved**)
    - you should also indicate where the evidence can be found if a '✓' is identified.
    - a X indicates that there is insufficient evidence to fully meet the criterion or it was not attempted.
- the total number of criteria achieved by the student for the unit.

You must be convinced, from the evidence presented, that students have worked independently to the required standard.

Your centre must internally standardise the assessment decisions for the cohort **before** you give feedback to students (see **Section 6.4.3**). When you are confident the internal assessment and standardisation process is complete, you can submit work for moderation at the relevant time. You **must not** add, amend or remove any work after it has been submitted to us for final moderation.

6.4.2 **Annotating students' work**

Each piece of NEA work must show how you are satisfied the assessment criteria have been met.

Comments on students' work and the Unit Recording Sheet (URS) provide a means of communication between teachers during internal standardisation, and with the OCR assessor if the work is part of the moderation sample.

6.4.3 **Internal standardisation**

It is important that all teachers are assessing work to common standards. For each unit, centres must make sure that internal standardisation of outcomes across teachers and teaching groups takes place using an appropriate procedure.

This can be done in a number of ways. In the first year, reference material and OCR training meetings will provide a basis for your centre's own standardisation. In following years, this, and/or your own centre's archive material, can be used. We advise you to hold preliminary meetings of staff involved to compare standards through cross-marking a small sample of work. After you have completed most of the assessment, a further meeting at which work is exchanged and discussed will help you make final adjustments.

If you are the only teacher in your centre assessing these qualifications, we still advise you to make sure your assessment decisions are internally standardised by someone else in your centre. Ideally this person will have experience of these types of qualifications, for example someone who:

- is delivering a similar qualification in another subject.
- has relevant subject knowledge.

You must keep evidence of internal standardisation in the centre for the OCR assessor to see.

We have a **guide** to how internal standardisation can be approached on our website.

6.4.4 **Reattempting work to improve the grade before submitting marks to OCR**

As described in **Section 6.2**, **before** submitting a final outcome to us for external moderation, you can allow students to repeat any element of the assignment and rework their original evidence. We refer to this as a reattempt. A reattempt allows the student to reflect on **internal** feedback, and to improve their work. A reattempt is **not** an iterative process where students make small modifications through ongoing feedback to eventually achieve the desired outcome.
Any feedback **must** be noted by the teacher and a record of this kept in centre. We have provided a feedback form for this purpose, which can be found in the OCR website.

A reattempt must be done before submission for external moderation. When a student submits the work to you as final for external moderation, they cannot complete any further work on any aspect of it.

6.4.5 **Submitting outcomes**

When you have assessed the work and it has been internally standardised, outcomes can be submitted to us. For the purpose of submission, outcomes will be considered as 'marks'. You will submit the total number of criteria achieved for units as marks. You can find the key dates and timetables on our **website**.

There should be clear evidence that work has been attempted and some work produced. If a student does not submit any work for an NEA unit, the student should be identified as being absent from that unit.

If a student completes any work at all for an NEA unit, you must assess the work using the assessment criteria and award the appropriate number of criteria. This might be zero.

6.4.6 **Resubmitting moderated work to OCR to improve the grade**

We use the term 'resubmission' when referring to student work that has previously been submitted to OCR for moderation. Following OCR moderation, if you and the student feel they have not performed at their best during the assessment, the student can, with your agreement, improve their work and resubmit it to you again for assessment. You must be sure it is in the student's best interests to resubmit the work for assessment. There is one resubmission opportunity per NEA assignment.

Students can only resubmit work using the **same** assignment if the assignment is still live. The live assessment dates and intended cohort will be shown on the front cover of the assignment. We will not accept work based on an assignment that is no longer live.

If students wish to resubmit a unit after the live assessment date has passed, they must submit work using the new live assignment.

6.5 Moderating NEA units

The purpose of external moderation is to make sure that the standard of assessment is the same for all centres and that internal standardisation has taken place.

The administration pages of our **website** give full details about how to submit work for moderation.

This includes the deadline dates for entries and submission of marks. For moderation to happen, you must submit your marks by the deadline.

6.5.1 **Sample requests**

Once you have submitted your marks, we will tell you which work will be sampled as part of the moderation process. Samples will include work from across the range of students' attainment. Copies of students' work must be kept until after their qualifications have been awarded and any review of results or appeals processed.

Centres will receive the final outcomes of moderation when the provisional results are issued. Results reports will be available for you to access. More information about the reports that are available is on our website.

We need sample work to help us monitor standards. We might ask some centres to release work for this purpose. We will let you know as early as possible if we need this from you. We always appreciate your co-operation.

7 Administration

This section gives an overview of the processes involved in administering these qualifications. Some of the processes require you to submit something to OCR by a specific deadline. More information about the processes and deadlines involved at each stage is on our **administration pages**.

7.1 Assessment availability

There are two assessment opportunities available each year for the externally assessed units: one in January and one in June. Students can be entered for different units in different assessment series.

All students must take the exams at a set time on the same day in a series.

NEA assignments can be taken by students at any time during the live period shown on the front cover. It is important you use the set assignment that is released in the same calendar year as the new cohort starts to ensure that students have two years to use the assignment.

There are two windows each year to submit NEA outcomes. Submission of student outcomes will initiate the moderation visit by the OCR Assessor.

You must make unit entries for students before you can submit outcomes to request a visit. All dates relating to NEA moderation are on our administration pages.

Qualification certification is available at each results release date.

7.2 Collecting evidence of student performance to ensure resilience in the qualifications system

Regulators have published guidance on collecting evidence of student performance as part of long-term contingency arrangements to improve the resilience of the qualifications system. You should review and consider this guidance when delivering this qualification to students at your centre. For more detailed information on collecting of evidence of student performance please visit our website.

7.3 Equality Act information relating to Cambridge Advanced Nationals

The Cambridge Advanced Nationals require assessment of a broad range of skills and, as such, prepare students for further study and higher-level courses.

The Cambridge Advanced National qualifications have been reviewed to check if any of the competences required present a potential barrier to disabled students. If this was the case, the situation was reviewed again to make sure that such competences were included only where essential to the subject.

7.4 Accessibility

There can be adjustments to standard assessment arrangements based on the individual needs of students. It is important that you identify as early as possible if students have disabilities or particular difficulties that will put them at a disadvantage in the assessment situation and that you choose a qualification or adjustment that allows them to demonstrate attainment.

If a student requires access arrangements that need approval from us, you must use **Access arrangements (online)** to gain approval. You must select the appropriate qualification type(s) when you apply. Approval for GCSE or GCE applications alone does not extend to other

qualification types. You can select more than one qualification type when you make an application. For guidance or support please contact the **OCR Special Requirements Team**.

The responsibility for providing adjustments to assessment is shared between your centre and us. Please read the JCQ document **Access Arrangements and Reasonable Adjustments**.

If you have students who need a post-exam adjustment to reflect temporary illness, indisposition or injury when they took the assessment, please read the JCQ document **A guide to the special consideration process.**

If you think any aspect of these qualifications unfairly restricts access and progression, please email **Support@ocr.org.uk** or call our Customer Support Centre on **01223 553998**.

The following access arrangements are allowed for this specification:

Access arrangement	Type of assessment
Reader/Computer reader	All assessments
Scribes/Speech recognition technology	All assessments
Practical assistants	All assessments
Word processors	All assessments
Communication professional	All assessments
Language modifier	All assessments
Modified question paper	Timetabled exams
Extra time	All assessments with time limits

7.5 Requirements for making an entry

We provide information on key dates, timetables and how to submit marks on our **website**.

Your centre must be registered with us to make entries. We recommend that you apply to become a registered centre with us well in advance of making your first entries. Details on how to register with us are on our **website**.

It is essential that unit entry codes are stated in all correspondence with us.

7.5.1 **Making estimated unit entries**

Estimated entries are not needed for Cambridge Advanced National qualifications.

7.5.2 **Making final unit entries**

When you make an entry, you must state the unit entry codes and the component codes. Students submitting work must be entered for the appropriate unit entry code from the table below.

The short title for these Cambridge Advanced Nationals is CAN AAQ. This is the title that will be displayed on our secure website, **Teach Cambridge**, and some of our administrative documents.

You do **not** need to register your students first. **Individual unit entries should be made for each series in which you intend to submit or resubmit an NEA unit or sit an externally assessed examination**.

Make a certification entry using the overall qualification code (see **Section 7.5**) in the final series only.

Unit entry code	Component code	Assessment method	Unit titles
F160	01	Written paper	Fundamentals of application development
F161	01	Written paper	Developing application software
F162A	01	Visiting	Designing and communicating UX/UI solutions
F162B	02	Remote	Designing and communicating UX/UI solutions
F163A	01	Visiting	Game development
F163B	02	Remote	Game development
F164A	01	Visiting	Website development
F164B	02	Remote	Website development
F165A	01	Visiting	Immersive technology solution development
F165B	02	Remote	Immersive technology solution development
F166A	01	Visiting	Software development
F166B	02	Remote	Software development

7.6 Certification rules

You must enter students for qualification certification separately from unit assessment(s). If a certification entry is **not** made, no overall grade can be awarded. These are the qualifications that students should be entered for:

- OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Certificate) - certification code H029.
- OCR Level 3 Alternative Academic Qualification Cambridge Advanced National in Computing: Application Development (Extended Certificate) - certification code H129.

7.7 Unit and qualification resits

Students can resit each unit and the best result will be used to calculate the certification result.

Resit opportunities must be fair to all students and **not** give some students an unfair advantage over other students. For example, the student must not have direct guidance and support from the teacher in producing further evidence for NEA units. When resitting an NEA unit, students must submit new, amended or enhanced work, as detailed in the JCQ **Instructions for Conducting Coursework**.

When you arrange resit opportunities, you must make sure that you do not adversely affect other assessments being taken.

Arranging a resit opportunity is at the centre's discretion. Summative assessment series must not be used as a diagnostic tool and resits should only be planned if the student has taken full advantage of the first assessment opportunity and any formative assessment process.

7.8 Post-results services

A number of post-results services are available:

- Reviews of results - if you think there might be something wrong with a student's results, you may submit a review of marking or moderation.
- Missing and incomplete results - if an individual subject result for a student is missing, or the student has been omitted entirely from the results supplied you should use this service.
- Access to scripts - you can ask for access to marked scripts.
- Late certification - following the release of unit results, if you have not previously made a certification entry, you can make a late request, which is known as a **late certification**. This is a free service.

Please refer to the JCQ **Post-Results Services booklet** and the **OCR Administration page** for more guidance about action on the release of results.

For NEA units the enquiries on results process cannot be carried out for one individual student; the outcome of a review of moderation must apply to a centre's entire cohort.

Appendix A: Guidance for the production of electronic evidence

Structure for evidence

The NEA units in these qualifications are units F162 - F166. For each student, all the tasks together will form a portfolio of evidence, stored electronically. Evidence for each unit must be stored separately.

An NEA portfolio is a collection of folders and files containing the student's evidence. Folders should be organised in a structured way so that the evidence can be accessed easily by a teacher or OCR assessor. This structure is commonly known as a folder tree. It would be helpful if the location of particular evidence is made clear by naming each file and folder appropriately and by use of an index called 'Home Page'.

There should be a top-level folder detailing the student's centre number, OCR candidate number, surname and forename, together with the unit code (F162 – F166), so that the portfolio is clearly identified as the work of one student.

Each student's portfolio should be stored in a secure area on the centre's network. Before submitting the portfolio to OCR, the centre should add a folder to the folder tree containing the internal assessment and summary forms.

Data formats for evidence

It is necessary to save students' work using an appropriate file format to minimise software and hardware capability issues.

Students must use formats appropriate:

- to their evidence
- for viewing for assessment and moderation.

Formats must be open file formats or proprietary formats for which a downloadable reader or player is available. If a downloadable reader or player is not, the file format is **not** acceptable.

Evidence submitted is likely to be in the form of word-processed documents, presentation documents, digital photos and digital video.

All files submitted electronically must be in the formats listed on the following page. Where new formats become available that might be acceptable, we will give more guidance. It is the centre's responsibility to make sure that the electronic portfolios submitted for moderation are accessible to the OCR assessor and fully represent the evidence available for each student.

Standard file formats acceptable as evidence for the Cambridge Advanced Nationals are listed here.

File type	File format	Max file size*
Audio	.3g2 .3ga .aac .aiff .amr .m4a .m4b .m4p .mp3 .wav	25GB
Compression	.zip .zipx .rar .tar .tar .gz .tgz .7z .zipx .zz	25GB
Data	.xls .xlsx .mdb .accdb .xlsb	25GB
Document	.odt .pdf .rtf .txt .doc .docx .dotx .	25GB
Image	.jpg .png .jpeg .tif .jfif .gif .heic .psd .dox .pcx .bmp .wmf	25GB
Presentation	.ppt .pptx .pdf .gslides .pptm .odp .ink .potx .pub	25GB
Video	.3g2 .3gp .avi .flv .m4v .mkv .mov .mp4 .mp4v .wmp .wmv	25GB
Web	.wlmp .mts .mov-1 .mp4-1 .xspf .mod .mpg	25GB

If you are using **.pages** as a file type, please convert this to a .pdf prior to submission.

*max file size is only applicable if using our Submit for Assessment service.

Submit for Assessment is our secure web-based submission service. You can access Submit for Assessment on any laptop or desktop computer running Windows or macOS and a compatible browser. It supports the upload of files in the formats listed in the table above as long as they do not exceed the maximum file size. **Other file formats and folder structures can be uploaded within a compressed file format**.

When you view some types of files in our Submit for Assessment service, they will be streamed in your browser. It would help your OCR assessor or examiner if you could upload files in the format shown in the table below:

File type	File format	Chrome	Firefox
Audio	.mp3	Yes	Yes
Audio	.m4a	Yes	Yes
Audio	.aac	No	Yes
Document	.txt	Yes	Yes
Image	.png	Yes	Yes
Image	.jpg	Yes	Yes
Image	.jpeg	Yes	Yes
Image	.gif	Yes	Yes
Presentation	.pdf	Yes	Yes
Video	.mp4	Yes	Yes
Video	.mov	No	Yes
Video	.3gp	Yes	No
Video	.m4v	Yes	Yes
Web	.html	Yes	Yes
Web	.htm	Yes	Yes

Appendix B: Command Words

External assessment

The table below shows the command words that will be used in exam questions. This shows what we mean by the command word and how students should approach the question and understand its demand. Remember that the rest of the wording in the question is also important.

Command Word	Meaning
Analyse	• Separate or break down information into parts and identify their characteristics or elements • Explain the different elements of a topic or argument and make reasoned comments • Explain the impacts of actions using a logical chain of reasoning
Annotate	• Add information, for example, to a table, diagram or graph
Calculate	• Work out the numerical value. Show your working unless otherwise stated
Choose	• Select an answer from options given
Compare	• Give an account of the similarities and differences between two or more items or situations
Complete	• Add information, for example, to a table, diagram or graph to finish it
Describe	• Give an account that includes the relevant characteristics, qualities or events
Discuss (how/whether/etc)	• Present, analyse and evaluate relevant points (for example, for/against an argument) to make a reasoned judgement
Draw	• Produce a picture or diagram
Explain	• Give reasons for and/or causes of something • Make something clear by describing and/or giving information
Give examples	• Give relevant examples in the context of the question
Identify	• Name or provide factors or features from stimulus
Label	• Add information, for example, to a table, diagram or graph until it is final
Outline	• Give a short account or summary
State	• Give factors or features • Give short, factual answers

Non examined assessment (NEA)

The table shows the command words that will be used in the NEA assignments and/or assessment criteria.

Command Word	Meaning
Adapt	• Change to make suitable for a new use or purpose
Analyse	• Separate or break down information into parts and identify their characteristics or elements • Explain the different elements of a topic or argument and make reasoned comments • Explain the impacts of actions using a logical chain of reasoning
Assess	• Offer a reasoned judgement of the standard or quality of situations or skills. The reasoned judgement is informed by relevant facts
Calculate	• Work out the numerical value. Show your working unless otherwise stated
Classify	• Arrange in categories according to shared qualities or characteristics
Compare	• Give an account of the similarities and differences between two or more items, situations or actions
Conclude	• Judge or decide something
Describe	• Give an account that includes the relevant characteristics, qualities or events
Discuss (how/whether/etc)	• Present, analyse and evaluate relevant points (for example, for/against an argument) to make a reasoned judgement
Evaluate	• Make a reasoned qualitative judgement considering different factors and using available knowledge/experience
Examine	• To look at, inspect, or scrutinise carefully, or in detail
Explain	• Give reasons for and/or causes of something • Make something clear by describing and/or giving information
Interpret	• Translate information into recognisable form • Convey one's understanding to others, e.g. in a performance
Investigate	• Inquire into (a situation or problem)
Justify	• Give valid reasons for offering an opinion or reaching a conclusion
Research	• Do detailed study in order to discover (new) information or reach a (new) understanding
Summarise	• Express the most important facts or ideas about something in a short and clear form

We might also use other command words but these will be:

- commonly used words whose meaning will be made clear from the context in which they are used
- subject specific words drawn from the unit content.

Some aspects of this document may not be fully accessible. If you have any problems with the accessibility of this format, please contact us.

Contact the team at:

01223 553998
computerscience@ocr.org.uk
ocr.org.uk
facebook.com/ocrexams
@OCR_ict
instagram.com/ocrexaminations
linkedin.com/company/ocr
youtube.com/ocrexams

To stay up to date with all the relevant news about our qualifications, register for email updates at **ocr.org.uk/updates**

Visit our Online Support Centre at **support.ocr.org.uk**

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT