Qualification Accredited

OCR
Oxford Cambridge and RSA

A LEVEL

Exemplar Candidate Work

COMPUTER SCIENCE

H446
For first teaching in 2015

H446/03/04 Summer 2019 series
Moderated components

Version 1

Contents

Would you prefer a Word version?

Did you know that you can save this pdf as a Word file using Acrobat Professional?

Simply click on **File > Save As Other . . .** and select **Microsoft Word**

(If you have opened this PDF in your browser you will need to save it first. Simply right click anywhere on the page and select ***Save as . . .*** to save the PDF. Then open the PDF in Acrobat Professional.)

If you do not have access to Acrobat Professional there are a number of **free** applications available that will also convert PDF to Word (search for *pdf to word converter*).

We value your feedback

We'd like to know your view on the resources we produce. By clicking on the icon above you will help us to ensure that our resources work for you.

Introduction

These exemplar answers have been chosen from the summer 2019 examination series.

OCR is open to a wide variety of approaches and all answers are considered on their merits. These exemplars, therefore, should not be seen as the only way to answer questions but they do illustrate how the mark scheme has been applied.

Please always refer to the specification https://www.ocr.org.uk/Images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf for full details of the assessment for this qualification. These exemplar answers should also be read in conjunction with the sample assessment materials and the June 2019 Examiners' report or Report to Centres available from Interchange https://interchange.ocr.org.uk

The question paper, mark scheme and any resource booklet(s) will be available on the OCR website from summer 2020. Until then, they are available on OCR Interchange (school exams officers will have a login for this and are able to set up teachers with specific logins – see the following link for further information https://www.ocr.org.uk/administration/support-and-tools/interchange/ managing-user-accounts/).

It is important to note that approaches to question setting and marking will remain consistent. At the same time OCR reviews all its qualifications annually and may make small adjustments to improve the performance of its assessments. We will let you know of any substantive changes.

Exemplar 1
Computational Methods

Computational methods will be vital to meet the aims of this project. The key method in this project is abstraction, but others will undoubtedly be useful.

Method	Justification of use	How it can be incorporated
Abstraction	To make best use of hardware resources, we must not include any features that are not essential to the program as they will only put more strain on the system. An abstract model will serve as a representation of reality that functions in only the ways that we need it to.	Making the game in 2 dimensions is a form of abstraction as it takes away the 3rd dimension, which is not essential to this kind of gameplay, but still makes for a good representation of reality and real-life situations. Making a game in 3D requires far more lines of code, time, storage and – _critically – _it requires more processing power client-side. Characters will use simple sprites and may not be 100% biologically correct – _they may not have visible fingers. On top of that, irrelevant environmental features can be discounted such as plants and animals.
Decomposition	Problems can be complex. Decomposition breaks down a complex problem into several simple ones, which are approachable and manageable. This makes things much easier on the developer's end.	Instead of looking at the game as a whole, we can look at the game as a set of its components, for example the surface, player controls and movement, bullet variety, and so on. We can put these components in order of importance (i.e. the surface and the controls are more important than the individual characters) and create them accordingly.
Logic, branches and loops	This refers to branching and loops. This adds interactivity, interest and complexity to the program. Without these logical operators the program is completely linear, but with them the program can produce different outcomes based on conditions or repeat instructions several times.	If/else statements must be used for the game to respond to the user's input, whether it be with the keyboard or mouse, as the program must be able to produce an outcome depending on the user's action. For example, if the left arrow key is depressed, the player should move left. Loops will be required to constantly refresh the surface

		for the game to run in real-time. They will also be necessary as part of algorithms such as score calculation. The game will have a main loop: a loop that runs once per frame. This will constantly check for states of the keys, to see if they are depressed or not and respond accordingly (e.g. move in a direction).
Pipelining and thinking concurrently	You don't want to have to wait for _something to finish, before starting another task. These techniques allow things to happen at the same time, or more specifically in the case of pipelining, allow a procedure to take place while waiting for the next input.	As a game runs at a constant frame rate, it cannot stop. That means it cannot halt the program to wait for a procedure to finish – _it must continue to run. This is where thinking concurrently comes in as the program must constantly be detecting user input, checking for collision events, calculating score, etc.
Performance Modelling	It is imperative that a game run smoothly so that there isn't delay between the player's input and what appears on screen. This phenomenon, 'lag', can make a game very cumbersome to play. Performance modelling will help to minimise lag by analysing what may cause it before implementing it into the final prototype.	Mathematical approximations for the runtime of algorithms can be used to predict potential performance issues and optimise wherever possible. I will use Big O notation to measure the runtime of my algorithms. Performance modelling can be applied iteratively by making several prototypes and identifying what is causing performance issues and optimise it for the next prototype.
Thinking ahead	This refers to careful planning to make sure the project is completed efficiently and in time. This will be useful as I will know how much time to allocate to each part of the project, and by code optimisation I can considerably reduce the overall project time.	The game will have many reusable elements, so I will use an object-oriented approach. This means I will begin by making object for bullets, characters and enemies and then implement them as and when I need them, avoiding

Mark given

10/10 for analysis section

Examiner commentary

As part of the analysis candidates are expected to explain why a problem is solvable by computational methods. When explaining why to use a computer and the use of computer methods to solve the problem too many candidates simply provide generic descriptions of computer methods. It is important for links between the essential features of the problem and appropriate computational methods is established in the narrative. In this example the candidate has briefly but explicitly explained how these computer methods apply to the proposed solution.

The choice of project is important and must be able to produce a problem that is multi-faceted and modular. In all but the most exceptional cases it will require a graphical user interface and will require the candidate to produce a series of linked modules during development. Simple quiz games or databases using a web based front end do not always provide the necessary features to enable the candidate to demonstrate the full range of skills required for this unit.

Exemplar 2
Parts that I can apply to my solution

Parts that I can apply to my solution

From these existing examples I have extracted some interesting ideas that I want to incorporate into my solution. Most notably, the ease of asking the systems to determine a digit makes the solutions I researched very appealing to my stakeholders.

Microsoft's OneNote

OneOne is very easy to use, however, it is not a library that can be implemented in other programs as it is designed to be used directly within OneNote. The simplicity of OneNote is a desirable feature that I will implement into my library in the form of simple and easy library requests. My library equivilent of OneNote's "Ink to Text" will be something like "ClassifyDigits()".

Google's Cloud Vision API

Google's API is quick and easy to access, however it requires a connection to the internet in order to make requests. I definitely want to make my library fast and easy to use like the Google API as this is a must for my stakeholders. The benefit of my library is that it grants the ability to work offline, like my stakeholders require. APIs often require a lot of setup code before they can be used, but my library will not.

Iron's OCR Library

Iron's OCR library is a product that is very similar to the library I want to design and develop: it is simple, fast and accurate. The library features a very simple implementation where users simple define an object and call a "Read()" method with an image path. This is how I envision users making requests to my library so I want to design a similar system. On top of this, the library features a system for setting paramaters about the classification. This allows the user to fine tune the trade off between speed and accuracy of the classification. This is another feature I definitely want to include in my library, however, I may not be able to include as many paramaters as Iron's library.

After researching different OCR algorithms and their benefits/problems, I have come to a conclusion about which algorithm is best for me. I am going to use machine learning with a neural network as I think it will be the easiest to implement and most effective. The neural network, once trained, will only consist of the graph data and the interface code. This means it will be small and easy to implement into a library, which is what my stakeholders want. As well as this, neural networks are fast (once trained) since the procedure consists of only multiplying and adding values, which computers are good at. The performance of the neural network can be set exactly as I require by training it and analysing its success rate. This means that I can ensure that the algorithm is definitely 90% accurate as my stakeholders want. The best way of maximising the success rate is by testing and experimenting with the algorithm using different parameters and conditions. Other algorithms, like matrix matching, require a library of image matrices to be stored making their implementation much larger.

The Z-Distribution is definitely a much better algorithm to use for normalising the image data. It creates a more consistent distribution which will greatly improve the accuracy of my system. It is easy to implement and will not require many CPU cycles, keeping the speed of my library fast. The only benefit of using the Min-Max normalisation algorithm is its speed. Perhaps users, in the future, could trade off some accuracy for speed if required. However, for now, the Z-Distribution is the algorithm I will use.

Mark given

10/10 for analysis

Examiner commentary

As part of the analysis candidates are expected to research the problem looking at existing solutions to similar problems. The analysis of this research should lead to identification of features appropriate to the proposed solution and the narrative should show how the results of the research can be applied to solving that problem.

Many students complete some detailed research into existing solutions but fail to relate these to their proposed solution. It is important that the student justifies how the research has impacted upon the proposed approach to the solution. In this case the student has completed detailed and well-focussed research into four areas related to the problem and followed up each piece of research with explicit justification of how that applies to the proposed solution.

Exemplar 3

What my program should do	How to test this	Justification
Have an evolution function	Once the evolution parameters have exceeded 10, 50, and 100; the characters will progress to their next evolutionary stage	This will help unlock mini-games and progress the game. If it cannot do this, it will be difficult to judge when mini-games should be unlocked. Also, this function provides a sense of accomplishment to the player and without it, the user may not feel like the game has any point existing.
Have a pause function	When the left and right arrow keys are simultaneously pressed, a subroutine will be executed that will halt all other processes in the game until the right and left keys are pressed simultaneously again	This will allow a user to temporarily halt functions in the game without having to save and quit it. This means the user will not have to play the game non-stop without breaks.
Have a save function	The data in the game will be able to be exported as a .json file to be used as a save state.	This means that the user can save their game onto a secondary storage device so the computer can be turned off without data being lost.
Have a load function	The .json file that has been saved to can be loaded to completely restore all data from that save state exactly as it was when it was stored.	This means that the save file in the previous criteria can be reloaded and is not meaningless to implement.
Have at least 3 functioning mini-games	Have 3 mini-games where you can either receive 5-, 10-, 15-, 20- or 30-point rewards to be exchanged for items and food and will increase evolution parameters.	This will help determine which characters to evolve into, also it will allow the user to access unlockable features, which will reward players for their effort and skill.
Contain a shop function	Have a part of the game where points can be exchanged for additional content. Have at least 3 backgrounds as point rewards.	The items here are the unlockable features to reward players once they have earned enough points in the mini-games to exchange for the additional content.
Contain a feeding/ stat checking option	Have a part of the game where you must feed your pet at least one time in a given interval (a minute spent on the home screen) or the happiness level will go down. The average happiness level will be used to calculate whether the	This will be a part of the game that teaches kids responsibility, which is one of the main aims of the game. The caring for the pet will teach kids how to look after an animal other than themselves and will help them learn how

Mark given

10/10 for analysis

Examiner commentary

As part of the analysis candidates are expected to identify suitable measurable success criteria. In many cases the success criteria identified are far too generic often referring to ease of use or aesthetic considerations. The success criteria will be used together with evidence of testing to establish the effectiveness of the solution. It is important the student identifies success criteria that can be definitively tested and used to determine the success, or otherwise, in developing that feature of the solution. The table above is a part of the success criteria for a game. The student identifies a measurable feature, establishes how that feature can be tested and provides justification for that feature as part of the success criteria.

Exemplar 4
Decomposing my problem

Decomposing my problem

I am going to use decomposition to break my system down into a series of smaller problems/programs. By doing this I can make the development process much easier to follow. In the section below I am going to look at the project as a whole and pick out the main sections that are suitable for decomposition. (I will design the algorithms and data structures in later sections as this section is purely for decomposition).

Decomposed section	Why I decomposed the section	A brief overview of the section
Neural network	This section consists of the neural network class and the data/methods it requires to function. The class is a vital part of the solution that will be frequently used in other sections of the system. Because of this, I have decided to decompose the data and algorithms that make up the class into their own section. This will ensure the class is ready for use (through testing) in other sections before I move onto them.	The structure of the neural network and how I will represent it in memory is something I must consider. The structure must allow for large networks and a simple way of propagating inputs through the network. I must be able to save this structure into a file that can be loaded back up into a new neural network object. Backpropagation is the main algorithm involved in training a neural network. It is a highly mathematical process that requires logical/mathematical thinking. I am going to derive this process and incorporate it efficiently into the class. I will look for patterns in the equations so that I can attempt to pipeline this process.
Library utilities	This part of the library will consist of a variety of tools which make the manipulation of data easier in other sections. By decomposing these tools into a section of their own, I can reduce duplication of code in other sections. It will also make the entire project more consistent (less prone to errors) if I have a single place that defines the utility tools. All of the other sections will make use of the tools in this section, so it is essential that I decompose these problems into their own section.	The utility tools are going to have maths tools that will help with mathematical tasks in other classes. For example, normalising data would be a task done by a maths tool. Other tasks may include finding: mean, standard deviation, and the Z distribution of an array. There will be image manipulation tools for loading and manipulating images etc. It will also contain tools for cropping and normalising the relevant information (about digits) from images. I will use the imaging tools to manipulate images of digits into the format I want. This will require some mathematical thinking as I will be manipulating pixel data. There will be tools for saving/loading the state of neural networks. This will be important for the library since I will want to save the most optimal network for use in the final library.
Library interface	The library interface is where the majority of library-user interactions will take place. Decomposing it into its own section makes logical sense since it is independent from the rest of the inner workings of the library. It makes use of internal library functions in order to process a user input and produce an output. By developing it separately from the backend library functionality, I can ensure that it is intuitive and easy-to-use as my stakeholders require.	The interface will consist of functions that allow the user to setup the library and classify digits. The setup function will let the user initialise the library with the setup they require. For example, they may want to select which neural network they want to use, so they can get the best performance for their project. The digit classifier function(s) will take an image input and return the classification of the digit. They will allow the user to select a mode that suits the user's needs in regards to the accuracy and speed of classification. These methods will be contained within a class called "Classifier". This class will encapsulate the library interface functionality and data.
Neural network training program	The training program will be an external program that uses the libraries utilities to train a neural network. Since the training program is an external project, it definitely should be decomposed and made into its own distinct stage in the project.	The program will make use of the libraries utilities in order to train and save a neural network. The program will be console based meaning it will display training information in text form. A console based application is faster than a GUI based one and speed is important for training neural networks. The program will use the MNIST dataset (LeCun, Cortes, & Burges, n.d.) of 60,000 training examples. Each iteration will display the following: 1) The current iteration. 2) The delta time since the last iteration. 3) The percentage of the last 500 training examples that were correct. 4) The average loss (error) of the neural network. These values can tell me how well the neural network is perfroming allowing me to make a decision about whether the network is correct.

Mark given

15/15 for design

Examiner commentary

Candidates are expected to decompose the main problem into suitable sections in order to develop a modular solution section by section. In many cases this is simply represented by a chart with little comment on the constituent parts. In this example the student has identified the essential elements of the solution explaining and justifying each section. This student then continues to develop detailed algorithms that define the solution accurately and completely using a range of appropriate techniques and methods including the use of detailed flow diagrams and pseudocode.

Exemplar 5
Justification of classes and methods

PlayerPet	happy	Attribute: integer	Used to keep track of the pet's happiness for evolution.	When a file is loaded, it will be checked if this is an integer or not.	Needed to implement the tiered evolution function.
PlayerPet	name	Attribute: string	Used to store the pet's name.	When a file is loaded, it will be checked if this is a string or not.	Needed for the user to bond with their digital pet.
TweetWindow	tweetvalue	Attribute: string	Used to store the value of the text that is tweeted.	Uses str() to make the contents a string.	Needed to show what the person is tweeting if they choose to.
TweetWindow	tweetpic	Attribute: pygame object	Used to store the image that is tweeted.	This will be created in-game with inputs generated with no user input so needs no validation.	Needed to show what the person is tweeting if they choose to.
TweetWindow	tweet	Method	Tweets the tweetvalue and tweetpic.	None	Needed to implement the tweeting function.
Button	text	Attribute: string	Contains the text that the button should display.	None	Shows that pressing the button will tweet the information.
Button	checkIfClicked	Method	Checks to see if the button has been pressed.	None	Needed to check if the user wants to tweet the data.

Mark given

12/15 for design

Examiner commentary

Validation is often an afterthought with some simply implying validation during the development process or providing a general outline stating validation will be used. Many will include this in a table of variables to be used, which is perfectly acceptable. In this case the student has provided a list of classes and methods to be used within the solution and identified and necessary validation. The image shows just a small section of the complete list of classes and methods identified by this student.

Exemplar 6
Version 2 testing

The snippet below validates the input parameter for the back propagation method. It does this in the same way as the feed forward method.

```
// If the input size is not equal to the input layer size
if (expectedOutput.Length != layerSizes[layers.Length])
{
    // Handle the error
    Console.WriteLine($"Cannot back propagate because the size ({expectedOutput.Length}) of the expected output array is
not the same size ({layerSizes[layers.Length]}) as the output layer.");
    return;
}
```

After implementing the validation and XML documentation, I decided that the stage was ready for testing again.

Stage 1: Version 2 testing

During the second testing session, I only revisited tests that failed or ones that I was not happy with. Here are the test results from after testing the second version.

Testing what	Test data/program state	Expected result	Actual result	Success/Partial/Failure
Feed forward an input	1) Array of wrong length	1) An informative error message should be displayed.	1) An informative error was displayed.	Success
Back propagate an expected output	1) Array of wrong length	1) An informative error message should be displayed.	1) An informative error was displayed.	Success

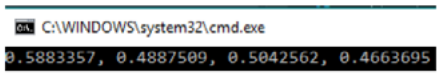
The screenshots below show evidence of the tests I performed on version 2 of the neural network.

Feed forward an input

I started by attempting invalid inputs in the "FeedForward()" method.

This time, the validation if statement was able to identify that the array length was incorrect.

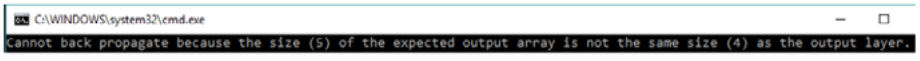
I decided to try a valid input in order to test whether the method still worked.



It still worked.

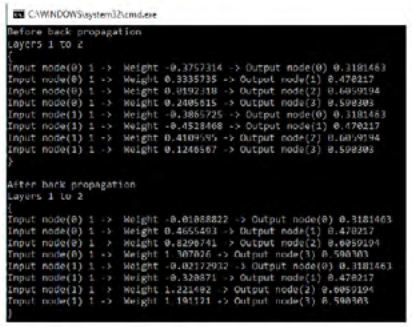
Back propagate an expected output

Next, I tried invalid inputs in the "BackPropagate()" method.



This time, the validation if statement was able to identify that the array length was incorrect.

I also retried a valid input to make sure the method was still working.



Stage 1: Review

What was done

I implemented the core functionality of the library – the neural network. This class is used in a lot of parts of the library meaning I was at a good stage in the development with a large chunk of the development done. I was able to set up a neural network that identified basic patterns in data. The neural network was able to successfully predict the outcome of unseen data. This was promising and I am excited to see how it performs with digits. The class I wrote was fast and easy to use which supports my success criteria. It handled validation perfectly by displaying any errors along with the parameters that caused them.

Stakeholder feedback

I contacted my stakeholder, Harry, and told him about the success of the first stage. He responded by saying he was impressed with the neural networks prediction ability and that he was very excited to see some digit classification using the network.

Success criteria I supported

Success criteria(s)	Justification
17	I have supported this success criteria by making sure I included detailed documentation on every class, method and attribute.
16	I have supported this success criteria by making sure I validate the inputs to methods. If there are any errors I make sure to print out an informative errors message that describes the issue.
18	The design and implementation of the neural network stage is modular in nature since it is easy to add and remove features without affecting the rest of the library.

Mark given

15/15 for Development and 10/10 for testing to inform development

Examiner commentary

Candidates are expected to provide evidence for each stage of the iterative process. It is expected that solutions will be modular and that each module is developed, tested and refined at each stage of the process. There should be justification for each test and evidence of test outcomes for each stage of the process. Candidates are expected to justify the choice of test and test data. They should also provide a commentary on the success or otherwise of the test and provide evidence of any actions taken following the test. This is a small example for secondary testing at the first stage of the development process to explain how issues identified through testing have been dealt with and re tested. The student relates all of the work to the success criteria and to the stakeholder's requirements.

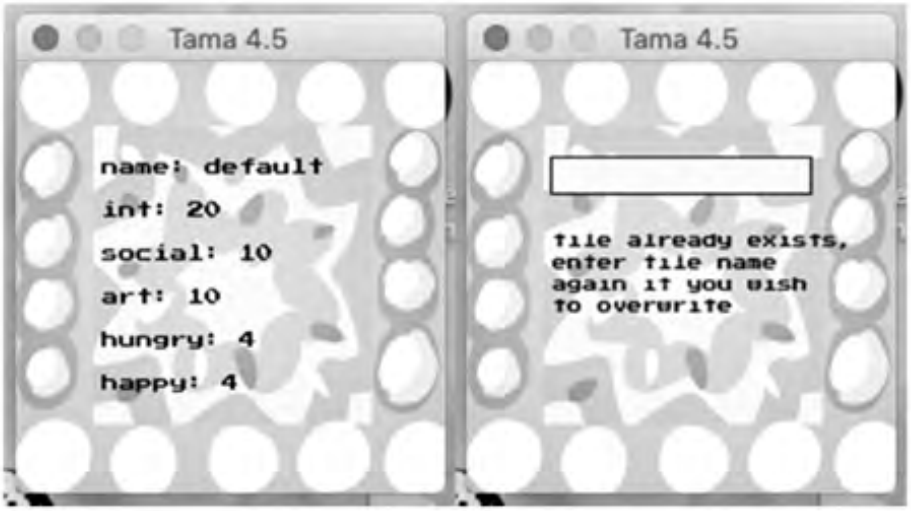
This student provides detailed evidence of testing at every stage of the development process. There is evidence of all failed tests and any remedial actions required with full justification for any actions taken. Many students rely on testing after the development process has been completed and do not provide sufficient evidence of how testing informed the process.

There must be clear evidence of testing within the development narrative including failed tests and remedial actions taken.

Exemplar 7
Usability testing

High contrast (3)	Any people who are playing the game and are colour-blind or partially sighted should be able to see the game clearly.	The game should have a contrast high enough to be able to discern what elements are shown when the image is in black and white.	Expected result occurred.	Yes, the implementation was successful. The images below clearly show that the greyscale versions of the game screenshots are easily understandable.	I may in the future decrease the saturation of the background, as the background is the most confusing part of the screenshots.
The image onscreen does not change too rapidly (4)	As part of my game, I have sprites moving erratically around the screen. If this occurred too fast, it may trigger seizures in children.	The home screen should not feature the pet moving faster than at 2 frames per second.	The pet moves around at 1.5 frames per second.	The implementation of this was successful. Unfortunately, since videos cannot be submitted in this format, I cannot display this.	No changes needed.
Simple language (5)	Some parts of this game require some words to make clear what is occurring in the game. These words will often be accompanied by pictures and/or will contain very simple language.	The text onscreen should either be accompanied by an image or the words in the phrases used must have a maximum of 3 syllables.	The expected result occurred.	The implementation of this was successful.	No changes needed.

(3)



(5)

Mark given

5/5 for testing to inform evaluation

Examiner commentary

Candidates are expected to test the usability of their system as part of the testing for evaluation. In many cases this is simply a report stating that users have seen the work and it is acceptable. In many cases candidates rely very heavily on a questionnaire and some form of rating system. These are all useful indications of user acceptance but do not demonstrate the usability features of the system. Evidence of testing and the outcomes for the usability features of the program are essential.

This student has identified a whole range of usability features and instigated suitable testing to demonstrate the effectiveness or otherwise of these features. The image shows just a small section of the test table and accompanying evidence. The full project narrative includes extensive evidence that usability features have been considered, justified and tested. This student also considered the views of stakeholders to provide complete evidence for the usability of the finished product.

As well as usability testing the evaluation testing should cover all the success criteria and show that the system is robust. In many cases the testing is limited to basic functionality without reference testing to show that the solution does not fall over easily under adverse conditions. This candidate included clear evidence of testing using a wide range of possible inputs and actions to show that the system was robust.

Exemplar 8

2. **The login interface should allow the user to enter their account details and gain access to the game.**

The login interface goes through the database that stores all the different account details and checked the username and corresponding password to access the game. The reading of the file is quite effective as all the details are stored to a list and has a quicker search method when looking for the username and password.

Tests:

***5, 6, 8,** the expected results stated that the user should be able to access the database and was tested to work while alerting the user if any details were incorrect.*

3. **The register interface should allow the user to select a username and password to create themselves an account. The program should ask the user to enter their password again to confirm and assign a recovery code for them.**

The simple connection to the csv and the format of the string allowed for a simple save method used to save all the details when the user entered the fields correctly. The recovery code connected the account in the csv and the user can get their details from the recovery interface if they forget their login details.

Tests:

***15, 16, 17, 19,** when the user goes to create the account the message box displaying were tested for the user to know exactly what detail is wrong. The recovery code was generated giving the uniqueness to each user account.*

4. **The recovery interface should allow the user to enter their code and access the information of their account.**

As the account is connected in a single line in the csv the data can be checked efficiently and can be retrieved quickly when the user enters the code in the recovery interface. The details are displayed on the screen for the user and can get to the login interface from the recovery menu to login once their details are shown. This makes it just clearer and easier to change and move onto logging in when the details are received.

Tests:

***22, 25,** the users details displayed when the correct recovery code was used showing that the database path was working too and could be used within account details frame.*

Mark given

14/15 for evaluation

Examiner commentary

Essentially the evaluation depends upon detailed and complete measurable success criteria being provided in the analysis. Evidence of testing during development and following development should be cross referenced against these success criteria and discussed.

This student has included detailed success criteria and systematically used evidence of testing to discuss the success, or otherwise, of the final product. The success criteria are clearly linked to the appropriate and relevant test evidence. This is an extensive section covering 5 full pages of evidence discussing each of the success criteria.

Did you know?

OCR Resources: *the small print*

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

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

OCR acknowledges the use of the following content: N/A

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

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

Looking for a resource?

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

www.ocr.org.uk/i-want-to/find-resources/

Need to get in touch?

If you ever have any questions about OCR qualifications or services (including administration, logistics and teaching) please feel free to get in touch with our **Customer Support Centre**.

General qualifications

Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk

www.ocr.org.uk