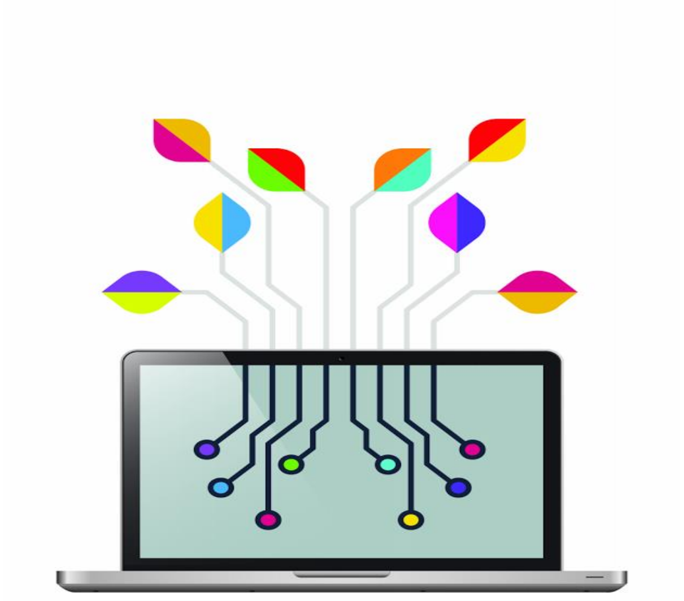
Examiners' Report/
Lead Examiner Feedback
Summer 2017

BTEC Level 3 Nationals in Computing
Unit 1: Principles of Computer Science
(31768H)



Edexcel and BTEC Qualifications

Edexcel and BTEC qualifications come from Pearson, the world's leading learning company. We provide a wide range of qualifications including academic, vocational, occupational and specific programmes for employers. For further information visit our qualifications websites at www.edexcel.com or www.btec.co.uk for our BTEC qualifications.
Alternatively, you can get in touch with us using the details on our contact us page at www.edexcel.com/contactus.

If you have any subject specific questions about this specification that require the help of a subject specialist, you can speak directly to the subject team at Pearson.
Their contact details can be found on this link: www.edexcel.com/teachingservices.

You can also use our online Ask the Expert service at www.edexcel.com/ask. You will need an Edexcel username and password to access this service.

Pearson: helping people progress, everywhere

Our aim is to help everyone progress in their lives through education. We believe in every kind of learning, for all kinds of people, wherever they are in the world. We've been involved in education for over 150 years, and by working across 70 countries, in 100 languages, we have built an international reputation for our commitment to high standards and raising achievement through innovation in education. Find out more about how we can help you and your learners at: www.pearson.com/uk

August 2017
Publications Code 31768H _1706_ER

Grade Boundaries

What is a grade boundary?

A grade boundary is where we set the level of achievement required to obtain a certain grade for the externally assessed unit. We set grade boundaries for each grade (Distinction, Merit, Pass and Near Pass). The grade awarded for each unit contributes proportionately to the overall qualification grade and each unit should always be viewed in the context of its impact on the whole qualification.

Setting grade boundaries

When we set grade boundaries, we look at the performance of every learner who took the assessment. When we can see the full picture of performance, our experts are then able to decide where best to place the grade boundaries – this means that they decide what the lowest possible mark should be for a particular grade.

When our experts set the grade boundaries, they make sure that learners receive grades which reflect their ability. Awarding grade boundaries is conducted to ensure learners achieve the grade they deserve to achieve, irrespective of variation in the external assessment.

Variations in external assessments

Each test we set asks different questions and may assess different parts of the unit content outlined in the specification. It would be unfair to learners if we set the same grade boundaries for each test, because then it would not take into account that a test might be slightly easier or more difficult than any other.

Grade boundaries for this, and all other papers, are on the website via this link: qualifications.pearson.com/gradeboundaries

Unit 1: Principles of Computer Science (31768H)

Grade	Unclassified	Near Pass	Pass	Merit	Distinction
Boundary Mark	0	12	24	39	54

Introduction

This was the first examination season for Level 3 BTEC Computing Unit 1 Principles of Computer Science 31768. This unit is assessed through a single written examination which is two hours in length and the number of marks available is 90. This unit is a mandatory unit for all learners studying the extended certificate, foundation diploma, diploma (all tech levels) and the extended diploma.

The examination for this unit will contain four sections and each section will have a scenario that will be used throughout the whole of that section. The scenario will be clearly stated at the beginning of each section. Each section is broken down into sub-questions which will then test learners on different areas of the specification and learners should are expected to apply their knowledge to the scenario.

Learners will be given an information booklet. They will be instructed to look at individual parts / sections of this during the examination in order to answer questions. The information booklet **may** give learners:

1. Information about problems that they need to solve.
2. Programming code for them to interpret, analyse or evaluate.
3. Requirements or designs for a new program that is needed.
4. An algorithm for them to interpret, analyse or evaluate.

At no point during the examination will learners be expected to write code in a particular language. Learners will only be given small pieces code to interpret, analyse or evaluate. Any code given to learners in the examination or information booklet will be written in C Family, Visual Basic, HTML 5 or Python 3.4.

All sections of the examination paper provide differentiation at all attainment levels and the paper is designed to be ramped in difficulty so that a larger percentage of higher grade marks are allocated to the later stages of the paper. This is a synoptic unit and therefore the skills and knowledge developed in this unit can be fed through all other units.

Introduction to the Overall Performance of the Unit

It was evident thought the range of responses that many learners had not covered the specification in full and subsequently were not fully prepared for the examination paper. This was particularly evident in questions that required learners to apply their knowledge. It is expected that learners will apply their knowledge to each scenario and respond with answers that are related to the scenario. However, in order to apply their knowledge, learners must be confident with the basic understanding or programming principles. It was also evident in questions that required learners to discuss, analyse or evaluate. Most learners were not able to meet the demands of these higher order command verbs which resulted in many learners achieving lower marks.

It is worth noting that the recommended amount of teaching time for this unit is 120 Guided Learners Hours (GLH). It is recommended that centres ensure that this amount of time is used to ensure that learners are equipped with the knowledge needed and to give learners enough time to allow them to apply their knowledge to different scenarios. This will therefore allow them to do this under exam conditions.

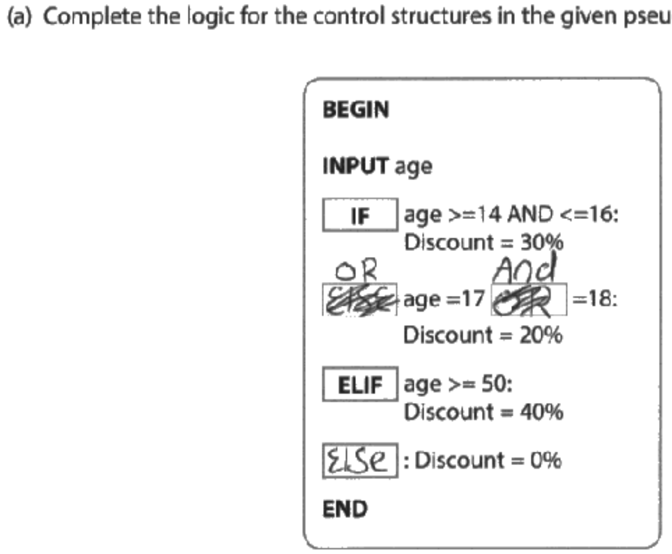
Centres are reminded that there is Sample Marked Learner Work (SMLW) available which is published on the Pearson website. This contains example responses based on questions from the Sample Assessment Materials (SAMs). Each question contains two responses; one good response, followed by a poor response. For each question the commentary includes the mark that each question would achieve and the reasons why.

Individual Questions

The following section considers each question on the paper, providing examples of learner responses and a brief commentary of why the responses gained the marks they did. This section should be considered with the live external assessment and corresponding mark scheme.

Question 1(a)

Most learners gained at least 2 marks for this question. Most learners were able to identify the correct two selection statements (IF and ELSE), however most learners were not able to identify the correct Boolean operator. Many learners incorrectly started AND instead of OR. This could possibly be because the AND operator was used previously in the pseudocode and therefore learners did not understand when each Boolean operator should be used.



(a) Complete the logic for the control structures in the given pseudocode.

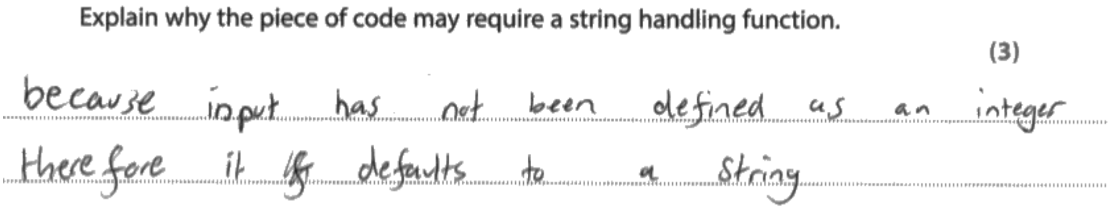
```
BEGIN
INPUT age
IF    age >=14 AND <=16:
          Discount = 30%
OR    And
~~Else~~ age =17 ~~OR~~ =18:
          Discount = 20%
ELIF  age >= 50:
          Discount = 40%
ElSe  : Discount = 0%
END
```

1 Mark given for:
'Else' in box 5
The learner has incorrectly placed a Boolean operator in box 2 rather than a selection statement. The learner has incorrectly used AND instead of OR in box 3.

Question 1(b)

The vast majority of learners answered this question incorrectly. Most learners argued that the input would need to be converted to a string as the user would type in letters or numbers. It is worth noting that learners were required to look at the programming code given and then apply their knowledge to that particular piece of code. Learners needed to state that the string handling function required was a string to integer so that the calculation / comparison could be made against the integers to avoid an error.

This question is an ‘Explain’ style question for 3 marks which suggests that one point needed to be stated with a further two linked expansions. Many learners did not provide expansions to the points they made which meant they limited their possible marks.

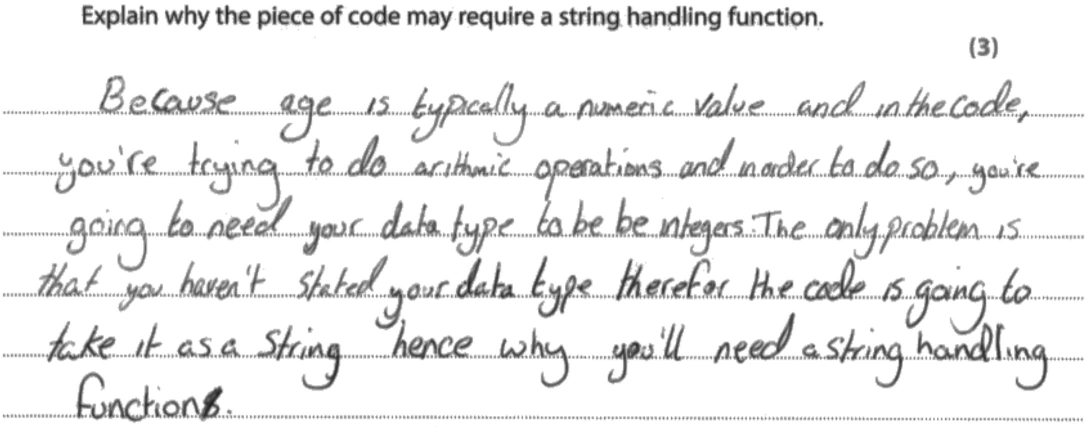


Explain why the piece of code may require a string handling function.

(3)

because input has not been defined as an integer therefore it defaults to a String

1 Mark would be given for:
‘because input has not been defined as an integer’ (1)



Explain why the piece of code may require a string handling function.

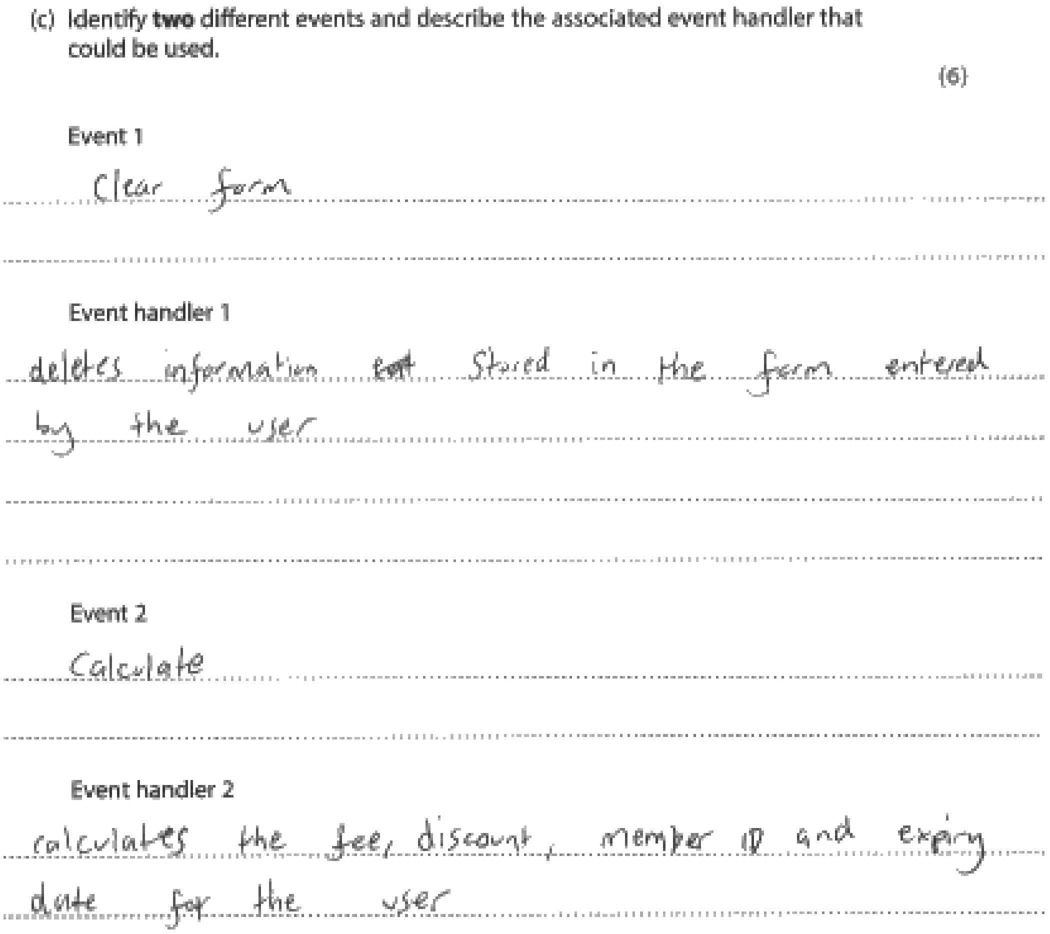
(3)

Because age is typically a numeric value and in the code, you're trying to do arithmic operations and norder to do so, you're going to need your data type to be be integers. The only problem is that you haven't stated your data type therefor the code is going to take it as a String hence why you'll need a string handling Function~~s~~.

2 Marks would be given for:
‘trying to do arithmetic operators’ (1)
‘You’re going to need your data type to be integers’ (1)

Question 1(c)

Most learners picked up some marks on this question. Most learners did attempt to state possible events but were not clear on the type of event. For example many learners wrote ‘calculation button’ which is too vague as it does not clearly state what the action is. A better answer would have been ‘Calculate button pressed’. Another example is ‘first name’ which does not state what the action is. A better answer would have been ‘First name clicked on’ or ‘First name clicked out/focus lost.’ A good answer could be ‘First name clicked out’ (1 mark) and then an event handler will run a data validation rule (1 mark) to ensure that data is present in the textbox (1 mark). It is worth noting that the answers in the mark scheme are only examples and other correct answers were awarded credit.
This question is an ‘Identify and Describe’ style question for 6 marks which suggests that two points should have been stated for different events with a further expansions of two points for each event for a further 4 marks. Many learners did manage to pick up some marks for correctly identifying a correct event handler but many lost marks for not correctly expanding this.



(c) Identify **two** different events and describe the associated event handler that could be used.

(6)

Event 1

Clear form

Event handler 1

deletes information ~~ent~~ stored in the form entered by the user

Event 2

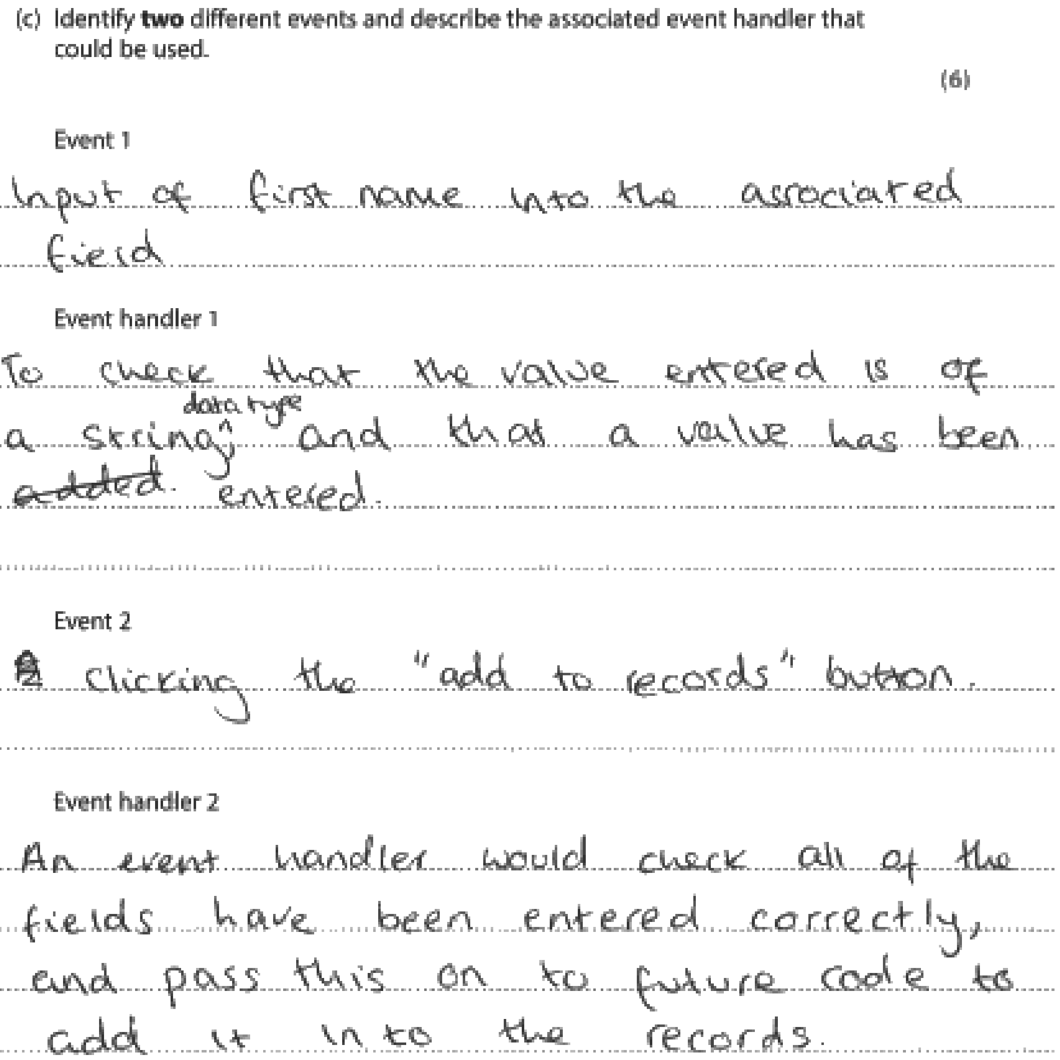
Calculate

Event handler 2

calculates the fee, discount, member ID and expiry date for the user

3 marks given for:
‘Clear form’ – this is too vague to give a mark here. Clear form button pressed would have been better.
‘Deletes information stored in the form entered by the user’ (1) (HANDLER)

‘Calculate’ – This is too vague.
‘Calculates the fee discount (1) (HANDLER) memberID and expiry data for the user’ (1) (HANDLER) - Just enough
The description for an event handler can achieve marks if a description of a process (eg a piece of code is run (1) to make a tick appear (1)) Or may be a description of multiple actions that appear on screen (eg form inputs cleared (1) ticks removed (1))



(c) Identify **two** different events and describe the associated event handler that could be used.

(6)

Event 1

Input of first name into the associated field

Event handler 1

To check that the value entered is of a string data type and that a value has been ~~added~~. entered.

Event 2

clicking the "add to records" button.

Event handler 2

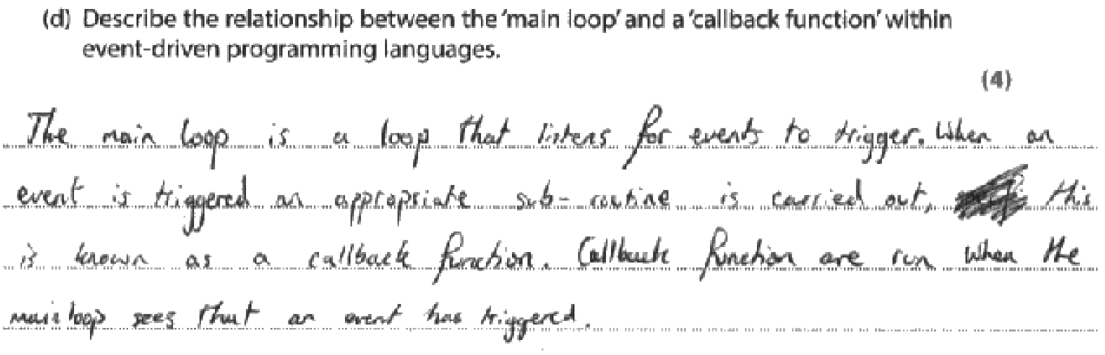
An event handler would check all of the fields have been entered correctly, and pass this on to future code to add it in to the records.

6 marks given for:
The examples in the MS are not an exhaustive list, you should use your professional judgement when giving marks for appropriate events and event handlers.
‘input of first name into the associated field (1) (KEYBOARD EVENT) ‘To check that the value entered is of a string data type (1) (HANDLER) and that a value has been entered (1) (HANDLER EXPANSION)

‘clicking the "add to records" button’ (1) (CLICK EVENT) ‘Check all of the fields have been entered correctly’ (1) (HANDLER) ‘and pass this on to future code to add it into the records’ (1) (HANDLER EXPANSION)
The learner has correctly stated two different events (data input and clicking) and described suitable handlers with suitable expansions. The event handlers relate to the events.

Question 1(d)

Some learners picked up some marks for this question with a lot of learners scoring 2 marks or more. Many learners were able to identify that the main loop will continually run while the program is running. A lot of learners did however confuse this with a general programming loop. Some learners were able to state what a callback function is. However very few learners were able to describe the relationship between the main loop and the callback function and how they work together.
This question is a ‘Describe’ style question for 4 marks which suggests that two linked points should have been stated for the main loop and two linked points should have been stated for the callback function. Although many learners picked up a mark for stating what a main loop and callback function was, very few learners picked up a second mark for a suitable expansion of each.

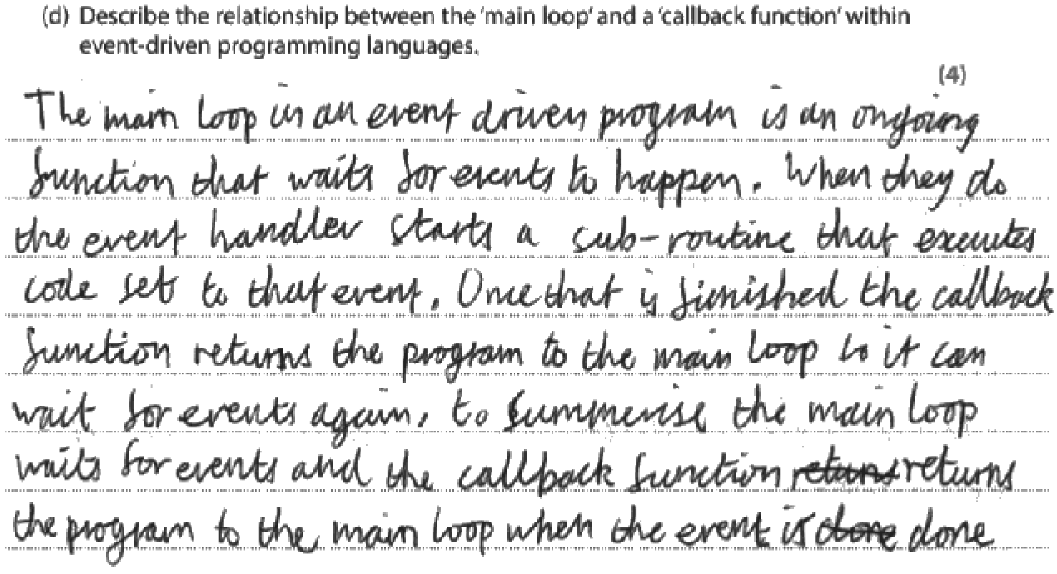


(d) Describe the relationship between the ‘main loop’ and a ‘callback function’ within event-driven programming languages.

(4)

The main loop is a loop that listens for events to trigger. When an event is triggered an appropriate sub-routine is carried out, this is known as a callback function. Callback function are run when the mai loop sees that an event has triggered.

2 marks given for:
‘main loop is a loop that lists for events to trigger’ (1) ‘when an event is triggered an appropriate sub-routine is carried out (1)
Final sentence is mark worthy but is a repeat of the first sentence.



(d) Describe the relationship between the 'main loop' and a 'callback function' within event-driven programming languages.

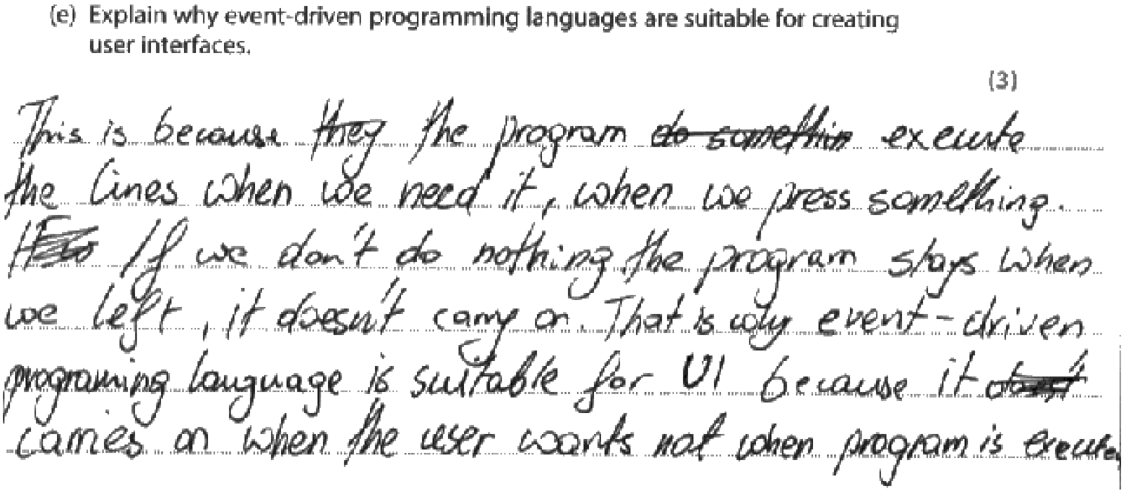
(4)

The main loop in an event driven program is an ongoing function that waits for events to happen. When they do the event handler starts a sub-routine that executes code set to that event. Once that is finished the callback function returns the program to the main loop so it can wait for events again, to summerise the main loop waits for events and the callback function ~~retuns~~ returns the program to the main loop when the event is ~~done~~ done

4 marks given for:
'Main loop…is an ongoing function (1) (enough for loops contonuously) that waits for events to happen' (1) 'when they do the event handler starts a sub-routine' (1) 'and then executes code set to that event' (1)
The learner has also made a valid point of 'returns the programme to the main loop' - however 4 marks have already been achieved

Question 1(e)

The vast majority of learners answered this question incorrectly. Many learners left the question blank or confused the context with another programming paradigm. Some learners were able to identify that event driven programming was required because interfaces have buttons / areas to click on or text boxes / areas to type into. However very few learners were able to take this further and give a further expansion.
This question is an ‘Explain’ style question for 3 marks which suggests that one point needed to be stated with a further two linked expansions. Although some learners were able to state a point and some managed to give a linked expansion, very few managed to give a second linked expansion for full marks.

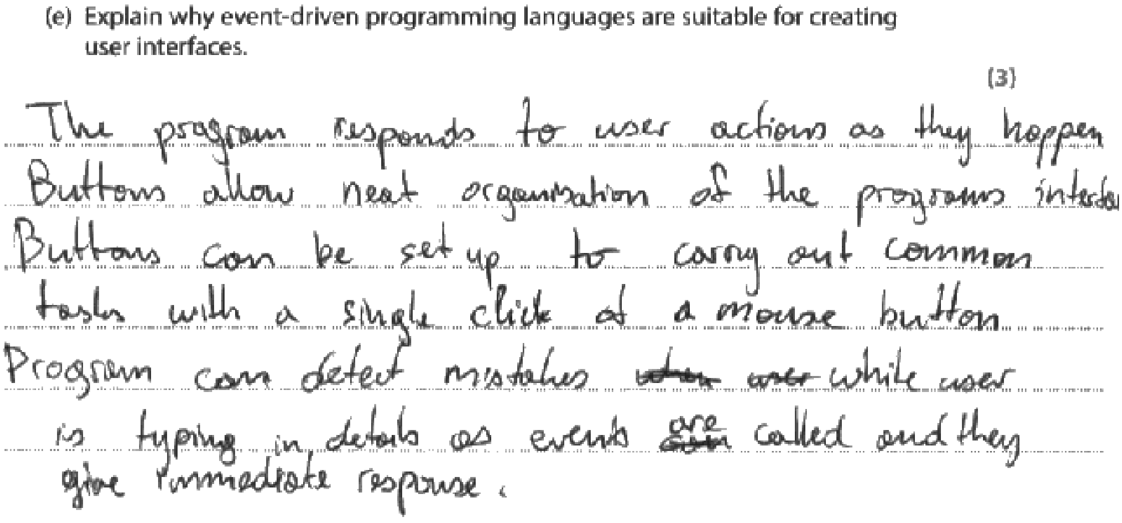


(e) Explain why event-driven programming languages are suitable for creating user interfaces.

(3)

This is because ~~they~~ the program ~~do somethin~~ execute the lines when we need it, when we press something. If we don't do nothing the program stays when we left, it doesn't carry on. That is why event-driven programing language is suitable for UI because it ~~dont~~ cames on when the user wants not when program is execute

2 marks given for:
‘The program execute the lines when we need it’ can be combined with ‘if we don’t do nothing the program stays when we left it (1) – there is enough here to show that the code responds to inputs.
when we press something (1) - just enough for 'such as mouse clicks'



(e) Explain why event-driven programming languages are suitable for creating user interfaces.

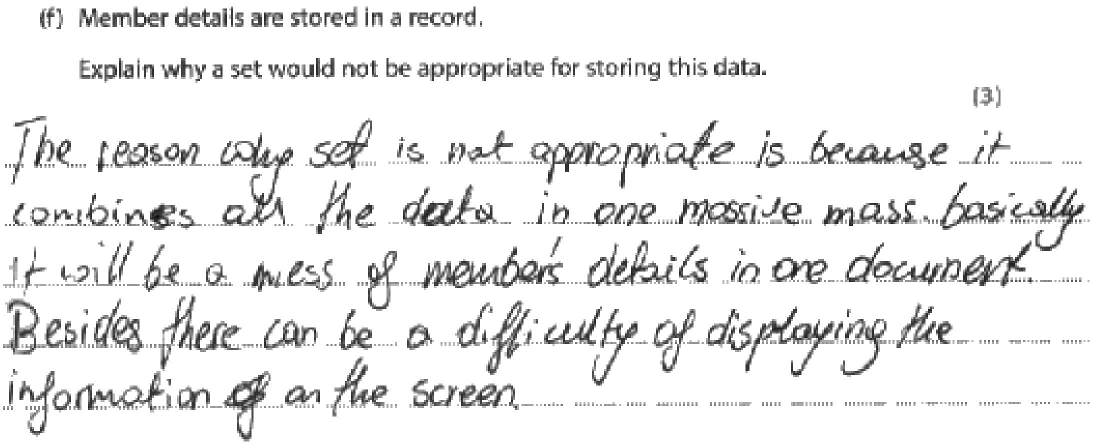
(3)

The program responds to user actions as they happen
Buttons allow neat organisation of the programs interfa
Buttons can be set up to carry out common tasks with a single click of a mouse button
Program can detect mistakes ~~when~~ ~~over~~ while user is typing in details as events ~~can~~ are called and they give immediate response.

3 marks given for:
‘The program responds to user actions as they happen’ (1) ‘to carryout out common tasks (1) with a single click of a mouse button (1)
The learner has correctly said ‘events are called and they give immediate response’ although this is a repeat of the first point and therefore credit cannot be given

Question 1(f)

The vast majority of learners answered this question incorrectly. A lot of learners did not know what a set was therefore leading to no credit being given. A lot of learners incorrectly focused their answers on sets being more secure and easier. It is worth noting that if learners explained why a record was more suitable, credit was still give given.
This question is an ‘Explain’ style question for 3 marks which suggests that one point needed to be stated with a further two linked expansions. Some learners managed to state a point such as sets only storing one data type or sets not storing duplicate data, however very few learners managed to expand this with a further two linked responses.



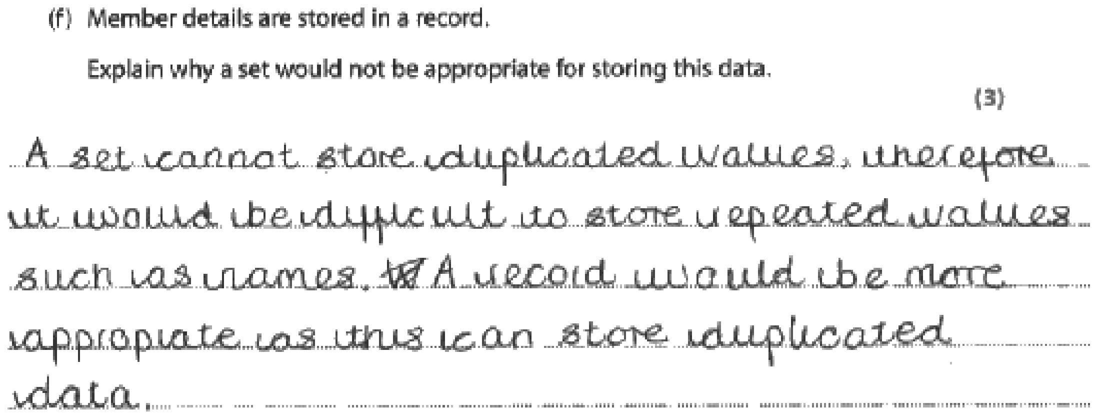
(f) Member details are stored in a record.

Explain why a set would not be appropriate for storing this data.

(3)

The reason why set is not appropriate is because it combines all the data in one massive mass. basically it will be a mess of member's details in one document. Besides there can be a difficulty of displaying the information ~~of~~ on the screen

1 mark given for:
'combines all the data in one massive mass. Basically it will be a mess' (1) as this shows just enough evidence to suggest that records will structure the data consistently



(f) Member details are stored in a record.

Explain why a set would not be appropriate for storing this data.

(3)

A set cannot store duplicated values, therefore it would be difficult to store repeated values such as names. A record would be more appropiate as this can store duplicated data.

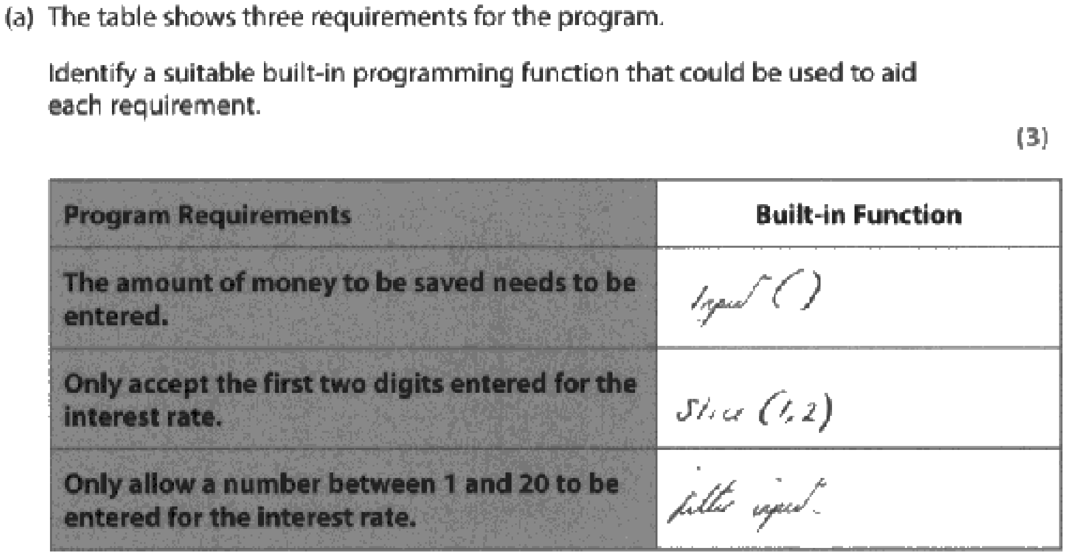
2 marks given for:

‘A set cannot store duplicated values’ (1) 'such as names' (1)

‘a record could be more appropriate as this can store duplicated values’ - although worded as an advantage of a record, it shows a clear understanding of the difference between sets and records and is worthy of a mark however this is a repeated point of the first marking point

Question 2(a)

Most learners gained 2 marks for this question. However, a large number of learners stated three different methods of data validation that could be used for each requirement rather than stating different built-in functions that could be used to meet the requirements.
Most learners were able to identify that an input function was required for requirement 1 and a range built-in function was required for requirement 2. However, very few learners were able to state that truncation was needed for requirement 2. It is worth noting that examiners also accepted other language specific built-in functions such as slice rather than truncation which is a built-in function in Python.



(a) The table shows three requirements for the program.

Identify a suitable built-in programming function that could be used to aid each requirement.

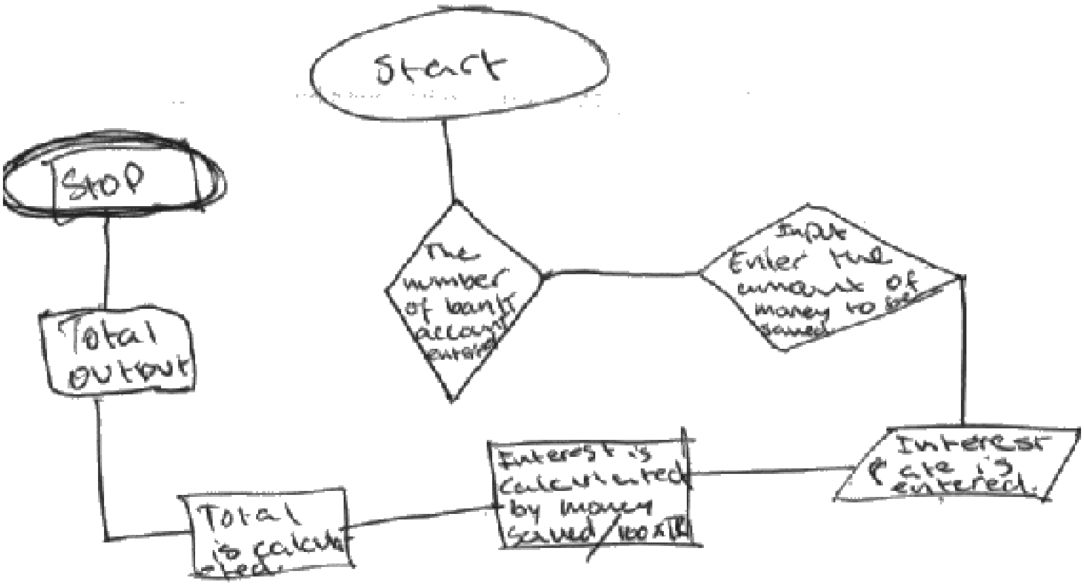
(3)

Program Requirements	Built-in Function
The amount of money to be saved needs to be entered.	Input ()
Only accept the first two digits entered for the interest rate.	Slice (1,2)
Only allow a number between 1 and 20 to be entered for the interest rate.	filter input.

2 marks given for:
'Input' in box 1 (1)
'Slice' in box 2 (1) – A function used for truncation in Python.

Question 2(b)

Most learners picked up some marks on this question with a lot of learners scoring 3 marks or more. Some learners copied the requirements that were given in the table into boxes which did not show sufficient understanding for credit to be given. It is worth noting that when drawing flowcharts, learners should use the correct BSC symbols for start/stop, input, output and process as stated in the specification. The vast majority of learners were able to identify correct variable names and use them consistently throughout the flowchart.
Most learners managed to solve a large part of the problem and followed the requirements given in the table. However the vast majority of learners struggled to use correct logic for the loop. Many learners either did not identify that a loop was required, how the loop would work (eg Comparisons = 0?) or incorrectly looped back to an incorrect position in their flowchart. It is worth noting that the example solution in the mark scheme is only an example and other correct ways of solving the problem were given credit.


Start
Stop
The number of bank account entered
Input Enter the amount of money to be saved
Total output
Interest rate is entered.
Interest is calculated by money saved/100 x IR
Total is calculated.

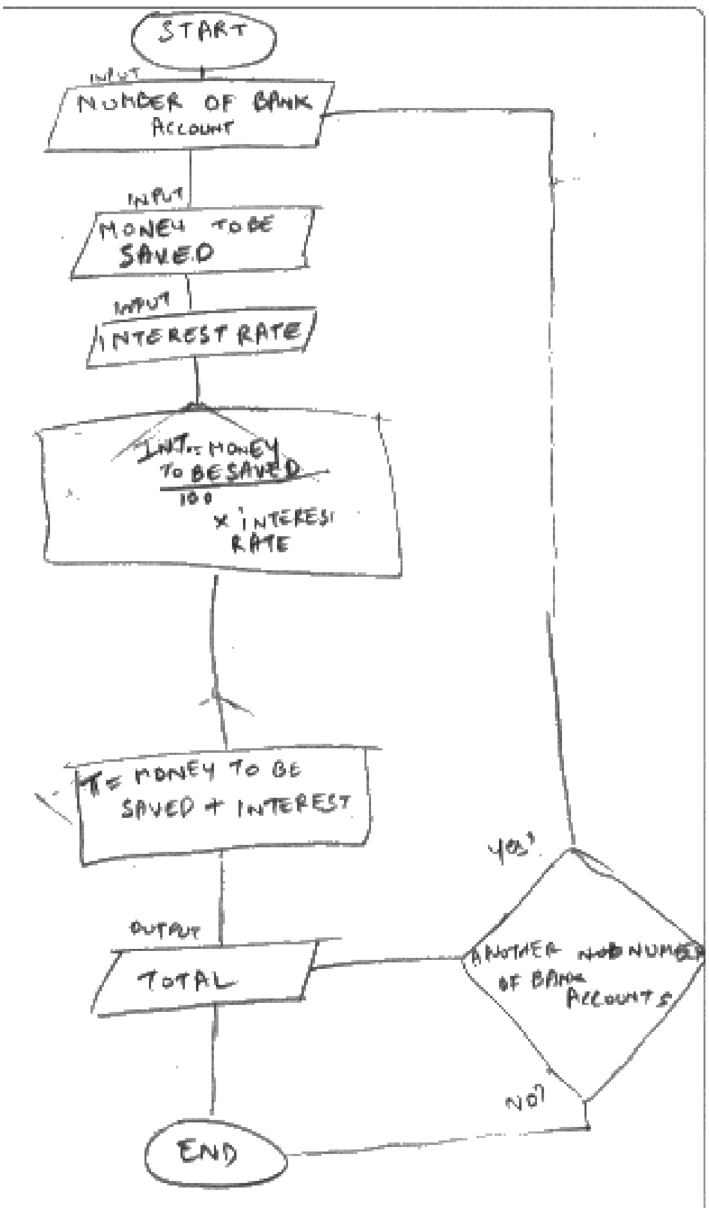
This solution meets the criteria for band 1 and 1 mark given

Structure - The structure of the flowchart only shows that the sequential instructions but makes no use of a loop. The flowchart does not follow standard structure and does not flow from top to bottom. Standard flow chart symbols have not been used correctly.

Names - The clarity is limited. This is because variable names are not clearly defined and therefore are not consistently used throughout.

Logical operations - Logical operations are not clear. For example 'Total is calculated' is not clear what logic is needed to actually create the total. No logic has been used to loop the code.

Overall - The learner has provided a partial solution to the problem. It is not a complete solution because they have used instructions to repeat certain parts of the flowchart for each account / shown how the total is calculated.


START
INPUT
NUMBER OF BANK ACCOUNT
INPUT
MONEY TO BE SAVED
INPUT
INTEREST RATE
~~INT.~~ = MONEY TO BE SAVED / 100 × INTEREST RATE
T = MONEY TO BE SAVED + INTEREST
OUTPUT
TOTAL
YES?
ANOTHER ~~NO~~ NUMBER OF BANK ACCOUNTS
NO?
END

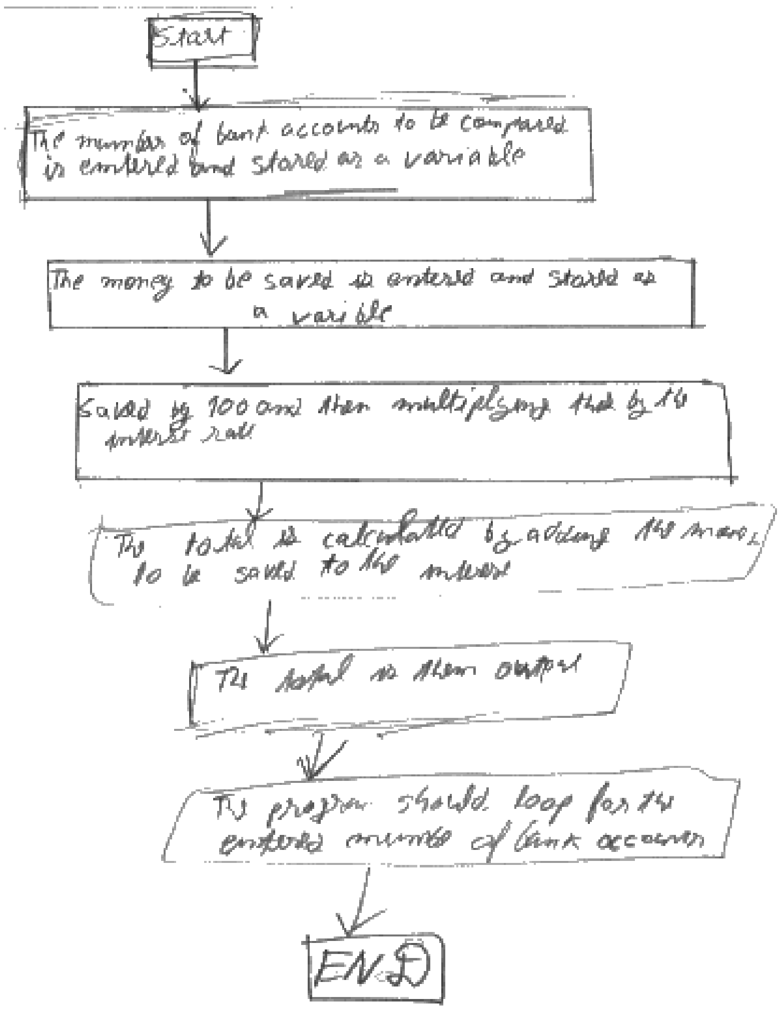
This solution meets the criteria for band 2 and 3 marks would be given.

Structure – The structure of the flowchart is most appropriate which is clear and easy to follow throughout. The flowchart is drawn from top to bottom.

Logical operations – Logical operations (eg 'Int = money to be saved / 100 * interest rate,' 'T = Money to be saved + interest' are accurate throughout.

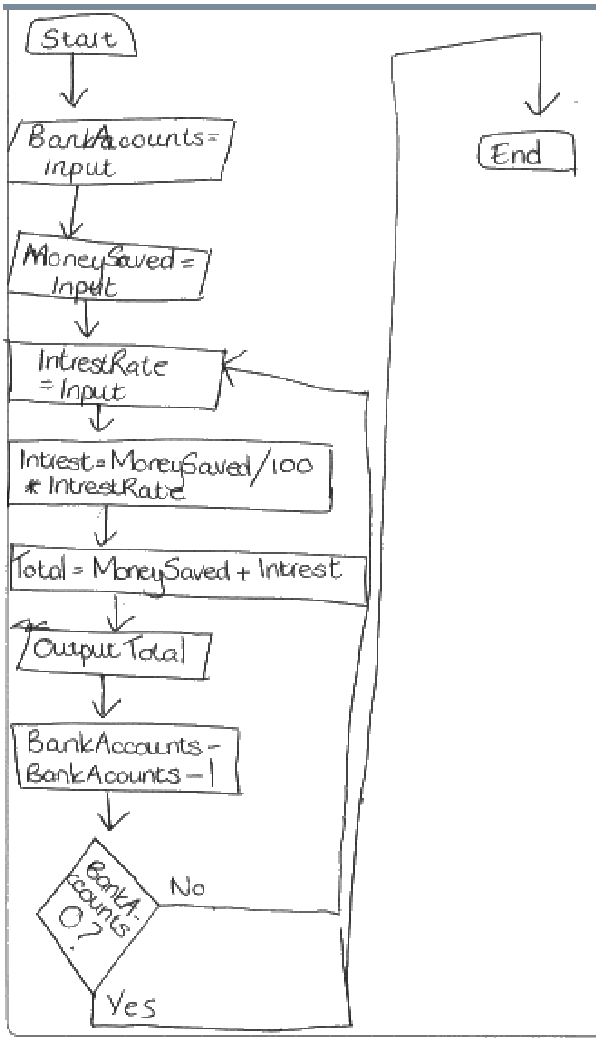
Names – Variable names are most appropriate although some are not (eg T) and are used inconsistently throughout the flowchart (eg T is used in one symbol but then referred to as Total in another).

Overall – The learner has provided an almost full solution to the problem and they have made correct use of a loop to create an efficient solution. It is however unclear however the logic that is used for when the loop would stop.


Start
The number of bank accounts to be compared is entered and stored as a variable
The money to be saved is entered and stored as a variable
Saled by 100 and then multiplying that by the interest rate
The total is calculated by adding the money to be saved to the interest
The total is then output
The program should loop for the entered number of bank accounts
END

No marks given

The learner has just copied the requirements provided and has not used any correct flowchart symbols or shown any logic.



5 marks given

Structure – The structure of the flowchart is appropriate which is clear and easy to follow throughout. The flowchart is drawn from top to bottom.

Names – Variable names are appropriate (eg InterestRate and Intrest) and are used consistently throughout the flowchart.

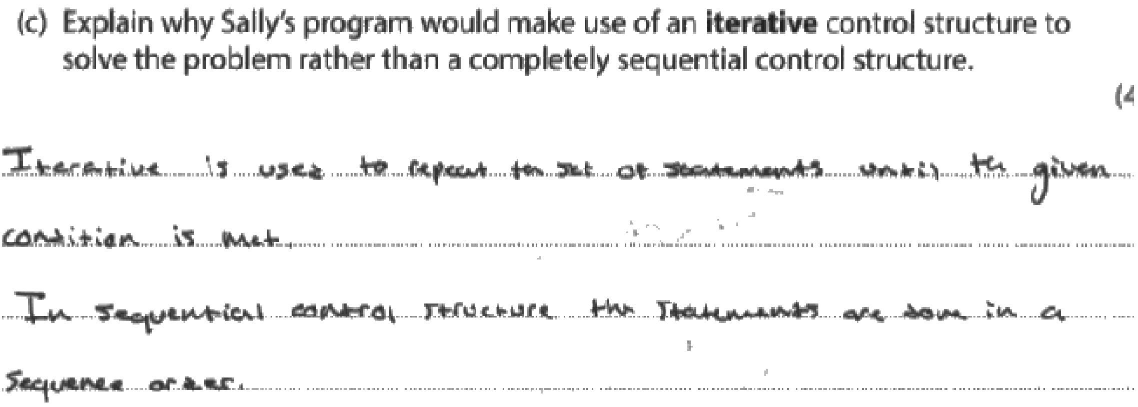
Logical operations– Logical operations (eg ‘BankAccounts 0?’ ‘intrest=MoneySaved/100*IntrestRate) are accurate in most places (minor error in terms of bank accounts- bank accounts-1).

Conventions – Conventions used are generally well known and suitable throughout.

Overall – The learner has provided a full solution to the problem however there is a minor error in the counter (bank accounts). They have made correct use of a loop to create an efficient solution. There is full clarity in the flowchart. Although there is an error in the counter a hollistic view is taken and as this is only a minor transcription error, their intention is clear, the solution is based in band 3.

Question 2(c)

The vast majority of learners answered this question incorrectly. A lot of learners did not know what iteration was which therefore prevented any credit being given. Most learners that did score marks were able to state that iteration is when code is looped for one mark or iteration makes the code more efficient for another mark. However to pick up further marks, learners needed to relate their answer to the scenario. Unfortunately, the vast majority of learners did not manage to relate to the scenario which therefore prevented a lot of learners scoring more than 2 marks.
This question is an ‘Explain’ style question for 4 marks which suggests that one point needed to be stated with a further three linked expansions. This question therefore demanded more depth from the learners. Some learners managed to state a point an initial point and every few manage to expand it. Hardly any learners were able to show more depth and get more than 2 marks for this question.



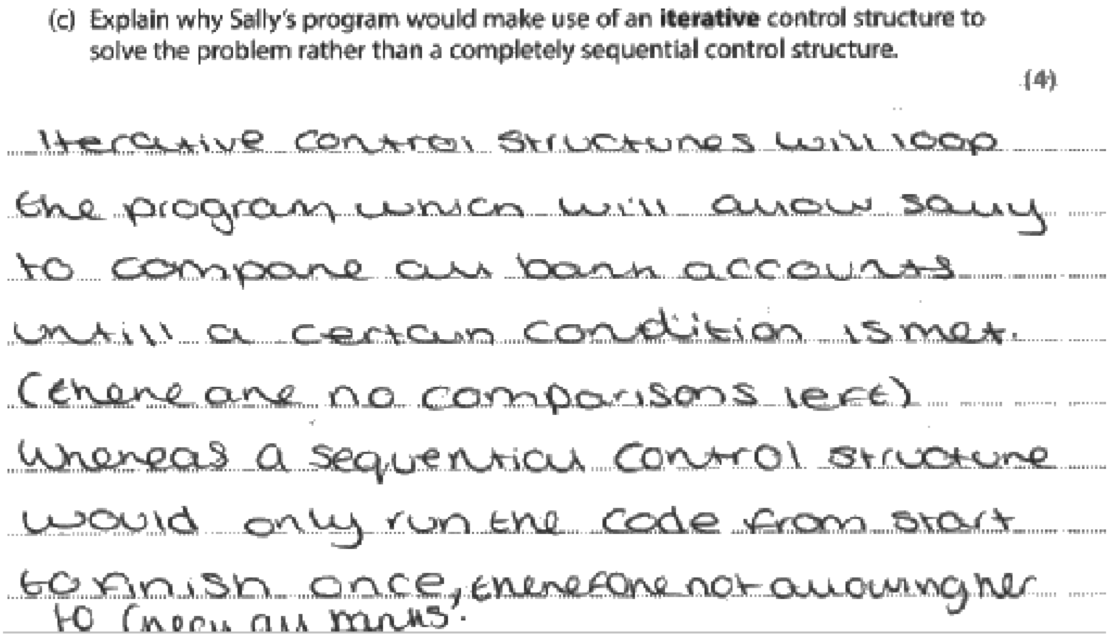
(c) Explain why Sally’s program would make use of an **iterative** control structure to solve the problem rather than a completely sequential control structure.

(4

Iterative is uses to repeat the set of statements until the given condition is met.

In sequential control structure the statements are done in a sequence order.

1 mark given for:
‘Repeat the set of statements’ (1)
'until the given condition is met' - not enough



(c) Explain why Sally's program would make use of an **iterative** control structure to solve the problem rather than a completely sequential control structure.

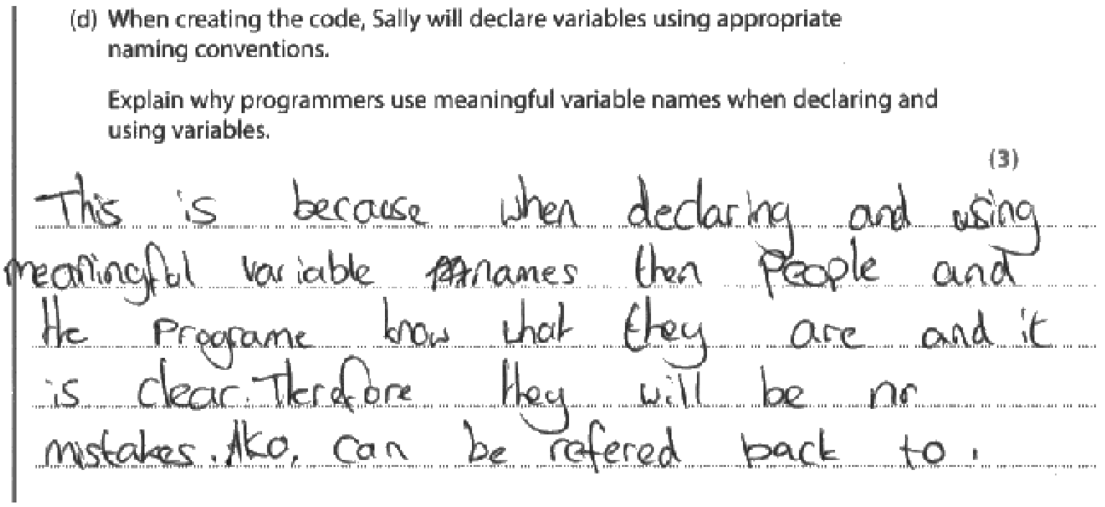
(4)

Iterative control structures will loop the program which will allow sally to compare all bank accounts untill a certain condition is met. (there are no comparisons left) Whereas a sequential control structure would only run the code from start to finish once, therefore not allowing her to check all marks.

2 marks given for:
'structures will loop' (1)
'allows sally to compare all bank accounts' (1)
'until a condition is met' - is not enough because it is not applied to the scenario

Question 2(d)

This question was very well answered by the vast majority of learners. Some learners unfortunately did not know what a variable was which resulted in no credit being given. Some learners focused on their answers on why naming conventions were needed rather than focusing their answer on why sensible variables names are important. Learners that scored marks generally tended to say that meaningful names ensure programmers know the purpose of the variable or other programmers will know what the variable means.
This question is an 'Explain' style question for 3 marks which suggests that one point needed to be stated with a further two linked expansions. Most learners did manage to state a suitable initial point and some learners did manage to provide a suitable expansion. Very few learners managed to show more depth and give a second linked expansion for full marks.



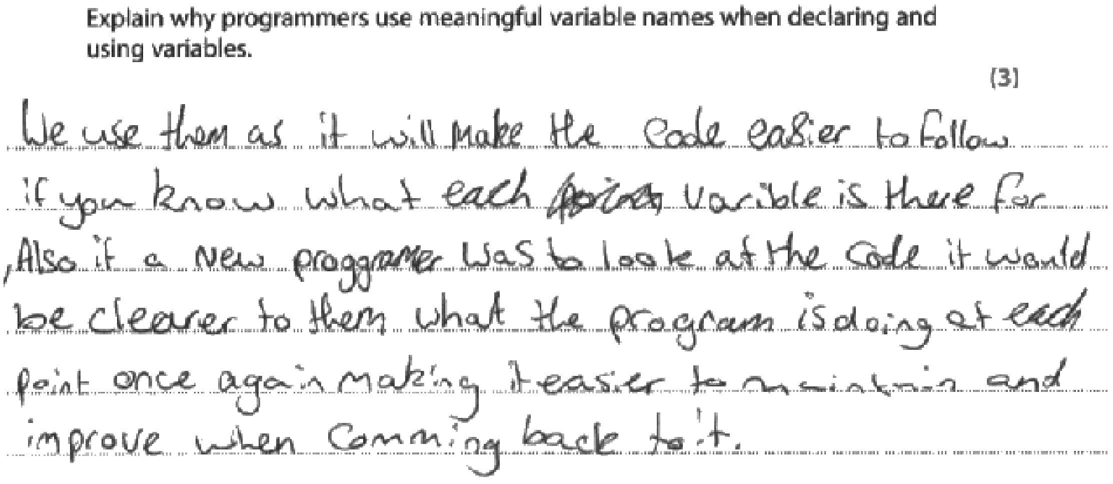
(d) When creating the code, Sally will declare variables using appropriate naming conventions.

Explain why programmers use meaningful variable names when declaring and using variables.

(3)

This is because when declaring and using meaningful variable names then People and the Programe know what they are and it is clear. Therefore they will be no mistakes. Also, can be refered back to.

1 mark given for:
'Therefore they will be no mistakes' (1) – There is just enough evidence here to show awareness of minimising errors.



Explain why programmers use meaningful variable names when declaring and using variables.

(3)

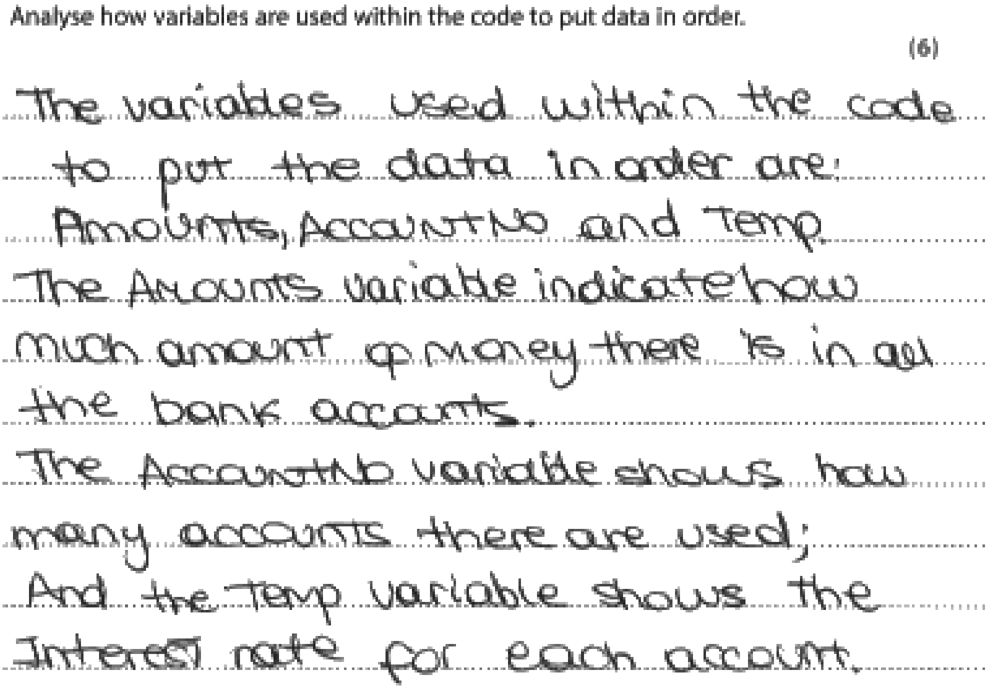
We use them as it will make the code easier to follow if you know what each varible is there for. Also if a new proggramer was to look at the code it would be clearer to them what the program is doing at each point once again making it easier to maintain and improve when comming back to it.

3 marks given for:

‘if you know what each variable is there for’ (1) this implies that the name implies its purpose.
‘Also if a new programmer…’ (1) this implies that others will understand the variable.
‘making it easier to maintain and improve when coming back to it’ (1)

Question 2(e)

This question was poorly answered by most learners. Unfortunately a lot of learners did not know what a bubble sort was and a lot of learners were not able to interpret pseudocode which prevented credit being given. A lot of learners who knew what a bubble sort was, simply described how a bubble sort works rather than how the variables in the given pseudocode sort the numbers into order. A lot of learners also focused on their answers on the WHILE and FOR Loops and the IF statement rather than how the variables in the given pseudocode sort the numbers into order.
Learners who scored marks within the first mark band were able to pick out some of the variables such as the Amounts, Account No and Temp and state their purpose but showed limited awareness of how they worked in the overall solution. Learners who got into the second and third mark band, managed analyse the pseudocode and backup their points with evidence from the given pseudocode code. Learners who scored higher marks tended to give more chains of reasoning and therefore manage to state a point and then take it further with more expansions.



Analyse how variables are used within the code to put data in order.

(6)

The variables used within the code
to put the data in order are:
Amounts, AccountNo and Temp
The Amounts variable indicate how
much amount of money there is in all
the bank accounts.
The AccountNo variable shows how
many accounts there are used;
And the Temp variable shows the
Interest rate for each account.

This response meets the criteria for band 1 and 1 mark given.

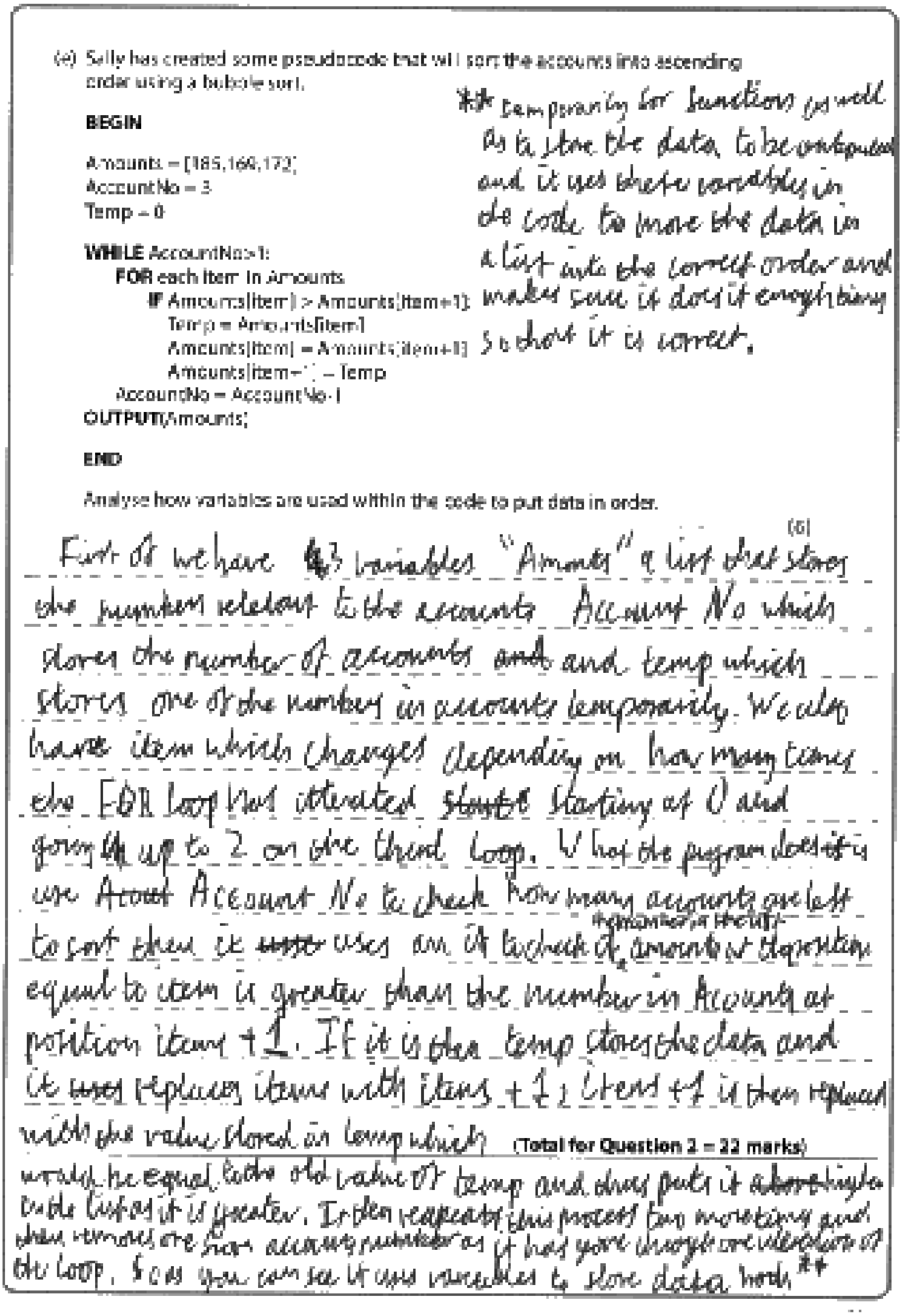
Technical vocabulary/Arguments – Some small points are made (eg. "Amounts Variable indicates amount of money," "AccountNo shows how many accounts") but this is very limited. Their description of the temp variable is inaccurate.

Application to question – The learner has managed to pick out some variable names such as Accounts, AccountNo and Temp. Their reasons to what they do are limited.

Balance – There is limited balance. Apart from stating a few variable names, there is no depth to their answers and to therefore their analysis is limited.

Chains of reasoning – No chains of reasoning are used. 'Amount variable indicates how much amount there is in the bank accounts' although the learner does not say why this is needed. "The AccountNo variable shows how many accounts there are used" although the learner does not say why this variable is needed.

Overall – Technical language is limited and does not support arguments. Chains of reasons are not made therefore leading to a general / superficial understanding of the question.



(e) Sally has created some pseudocode that will sort the accounts into ascending order using a bubble sort.

```
BEGIN

Amounts = [185,169,172]
AccountNo = 3
Temp = 0

WHILE AccountNo>1:
    FOR each item in Amounts
        IF Amounts[item] > Amounts[item+1]:
            Temp = Amounts[item]
            Amounts[item] = Amounts[item+1]
            Amounts[item+1] = Temp
    AccountNo = AccountNo-1
OUTPUT(Amounts);

END
```

Analyse how variables are used within the code to put data in order.

(6)

First of we have ~~4~~ 3 variables "Amounts" a list that stores the numbers relevant to the accounts Account No which stores the number of accounts ~~and~~ and temp which stores one of the numbers in accounts temporarily. We also have item which changes depending on how many times the FOR loop has iterated ~~start~~ starting at 0 and going ~~up~~ up to 2 on the third loop. What the program does is use ~~Acount~~ Account No to check how many accounts are left to sort then it ~~uses~~ uses an if to check if the number in the list amounts at the position equal to item is greater than the number in Accounts at position item +1. If it is then temp stores the data and it ~~uses~~ replaces items with items +1 & items +1 is then replaced with the value stored in temp which

(Total for Question 2 = 22 marks)

would be equal to the old value of temp and thus puts it ~~above~~ higher in the list as it is greater. It then repeats this process two more times and then removes one from account number as it has gone through one iteration of the loop. So as you can see it uses variables to store data both **

** temporarily for functions as well as to store the data to be outputted and it uses these variables in the code to move the data in a list into the correct order and makes sure it does it enough times so that it is correct.

This response meets the criteria for band 3 and 6 marks given

Technical vocabulary/Arguments – The learner has accurately described the first pass and then stated that this process would be repeated a further two times (for the three numbers). The learner has picked out different variables from the Pseudocode and then applied their own knowledge throughout the question.

Application to question – The learner has accurately used correct variable names from the pseudocode and used these to describe why they are needed.

Balance – The learner has covered a range of different variables and clearly described why they are needed in the bubble sort.

Chains of reasoning – Lots of chains of reasoning for example "item which changes" and then backed up with "depending on how many time the FOR loop has iterated." "What the program does is use the AccountNo to check how many accounts are left" and then backed this up with "and then it uses an if to check if the number in the list...." "If it is then temp stores the data it replaces items with items+1" and then continued to say "items+1 is then replaced with the value stored in temp."

Overall – Technical language is accurate and supports their arguments. The learner has described a wide range of points and uses chains of reasoning to show full awareness of what the variables are for and why they are important in bubble sorts. Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question.

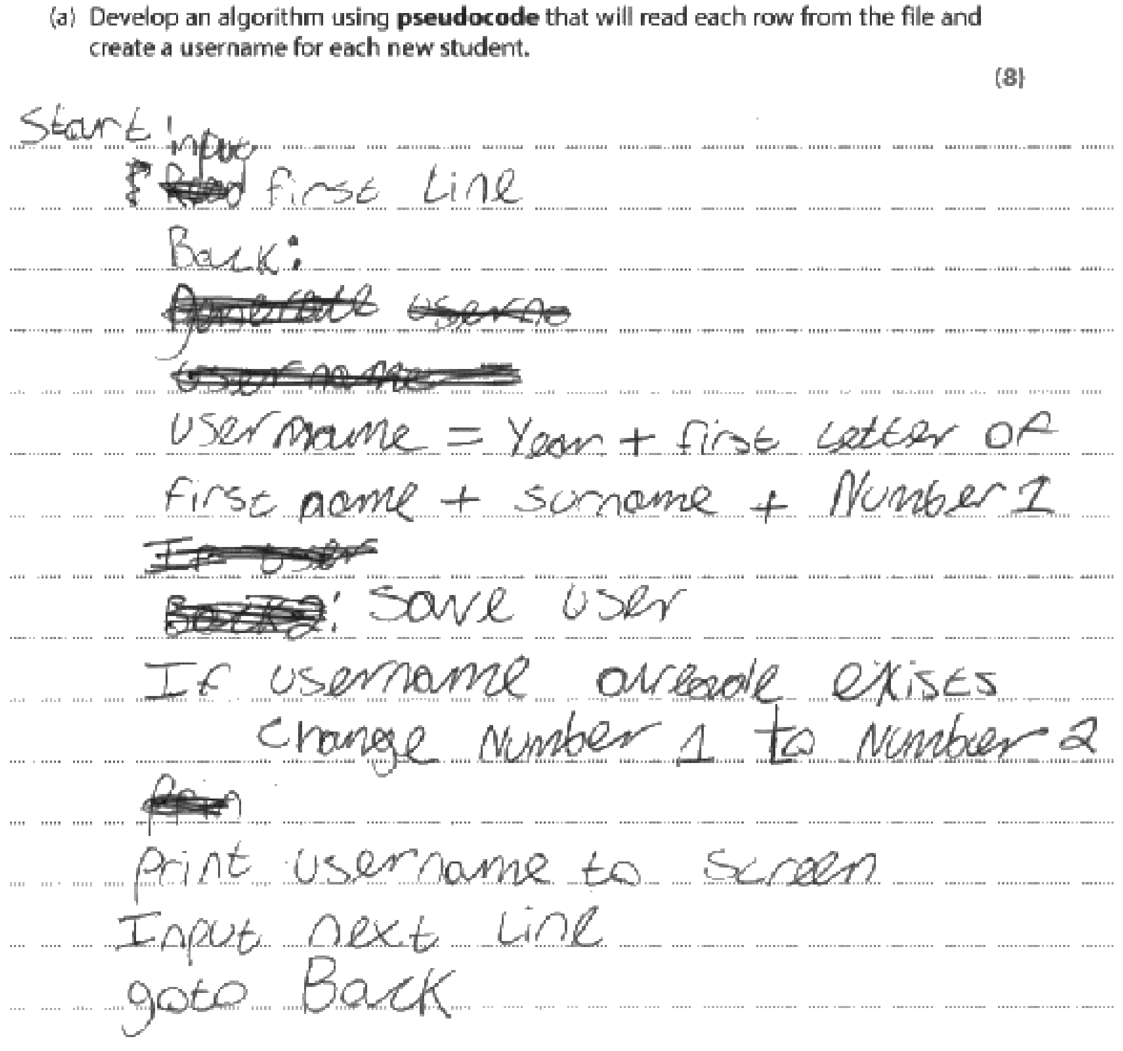
Question 3(a)

This question was poorly answered by most learners. The vast majority of learners did not know how to write pseudocode and therefore a lot of learners scored 0 marks. Some learners misread the question and created a username for each person rather than writing pseudocode that will create the usernames. Some learners focused their answer on writing pseudocode that produces a single username for a single person rather than iterating to produce a username for each person in the text file.

It is worth noting that examiners are aware that there are different variations in writing pseudocode and these are taken into account when marking pseudocode questions. Credit will be given for the correct use of logic. Learners were given steps to follow in the information booklet and those that got into mark band 2 or 3 managed to break these steps down further and then cover each step logically.

It is worth noting that the example solution in the mark scheme is only an example and other correct ways of solving the problem were given credit.

Learners would benefit from being more prepared to write pseudocode questions. It’s recommended that algorithms are taught alongside the whole unit which increase in difficulty overtime.



(a) Develop an algorithm using **pseudocode** that will read each row from the file and create a username for each new student.

(8)

```
Start: input
    ~~Read~~ first line
     Back:
     ~~Generate username~~
     ~~username~~
     username = Year + first letter of
     first name + surname + Number 1
    ~~If user~~
     ~~[illegible]~~: Save user
    If username alreade exists
        Change Number 1 to Number 2
    ~~[illegible]~~
    Print username to Screen
   Input next line
   goto Back
```

This solution meets the criteria for band 1 and 2 marks given.

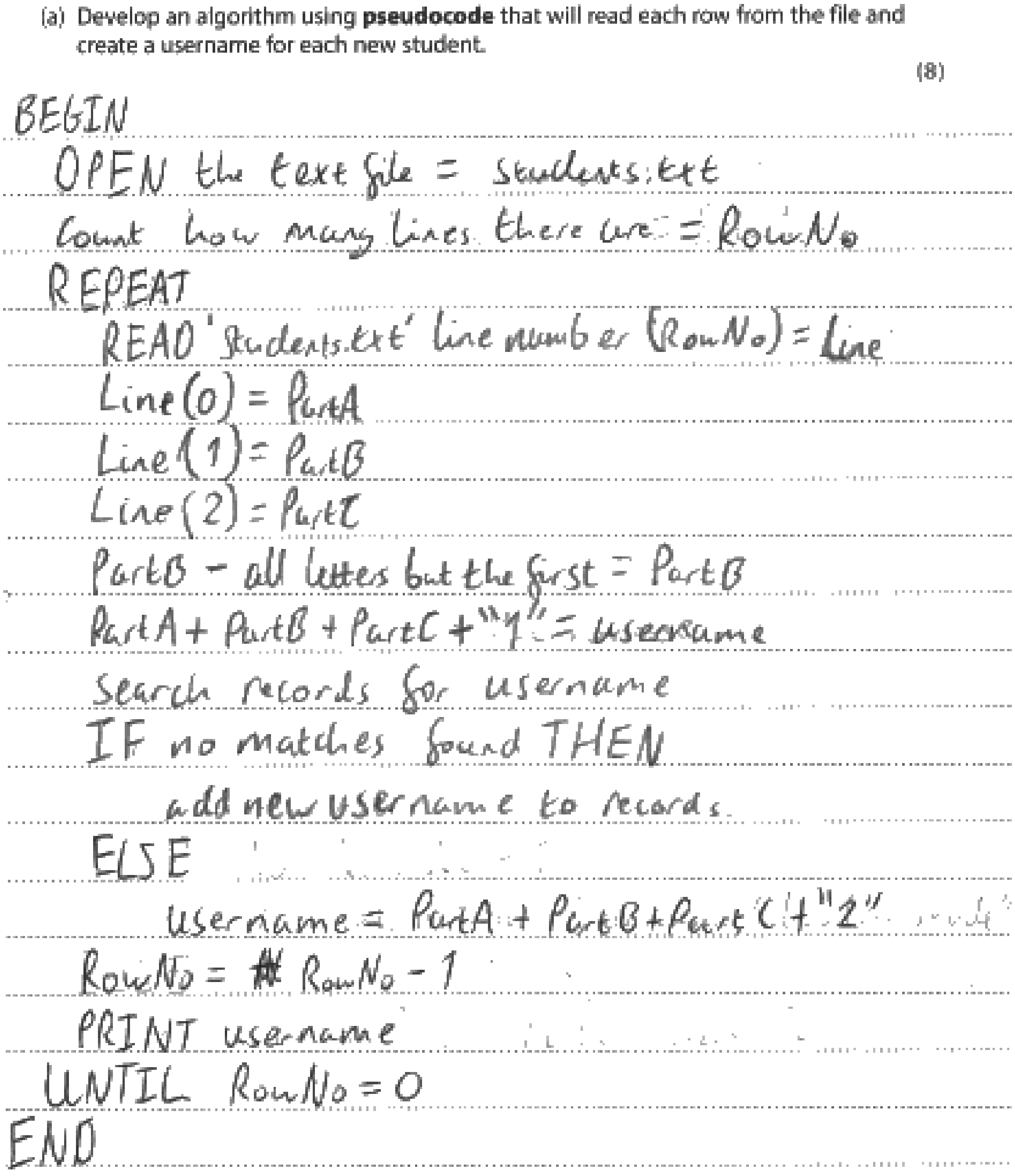
Structure – The algorithm is largely indented the whole with some use of hierarchy for the IF statement.

Names – The learner has not clearly defined variable/process names (e.g. username, year etc.). The learner has used concatenation to add the different parts of the username together however this is essentially just copied from the information booklet.

Logical operations – The learner has correctly identified that a loop will be needed and has correctly stated that the next is input and then looped again, how it is unclear when this loop would stop, they used an IF statement to check if the username already exists and then changed the number from 1 to 2, however no additional checks are made (and then to create this further). A while loop would have been better.

Conventions - Accepted conventions have not been used throughout. They have made use of begin, if, print, input etc., although these are not always necessary.

Overall – The learner has provided a partial solution to the problem. It is not complete because they have not used logic to separate the different parts of the text file into different variables/parts or correct logic to carry out further checks if the username is changed to ensure it is unique.



(a) Develop an algorithm using **pseudocode** that will read each row from the file and create a username for each new student.

(8)

```
BEGIN
  OPEN the text file = students.txt
  Count how many lines there are = RowNo
  REPEAT
    READ 'students.txt' line number (RowNo) = Line
    Line(0) = PartA
    Line(1) = PartB
    Line(2) = PartC
    PartB - all letters but the first = PartB
    PartA + PartB + PartC + "1" = username
    Search records for username
    IF no matches found THEN
        add new username to records.
    ELSE
        username = PartA + PartB + PartC + "2"
    RowNo = ~~#~~ RowNo - 1
    PRINT username
  UNTIL RowNo = 0
END
```

This solution meets the criteria for band 2 and 5 marks given.

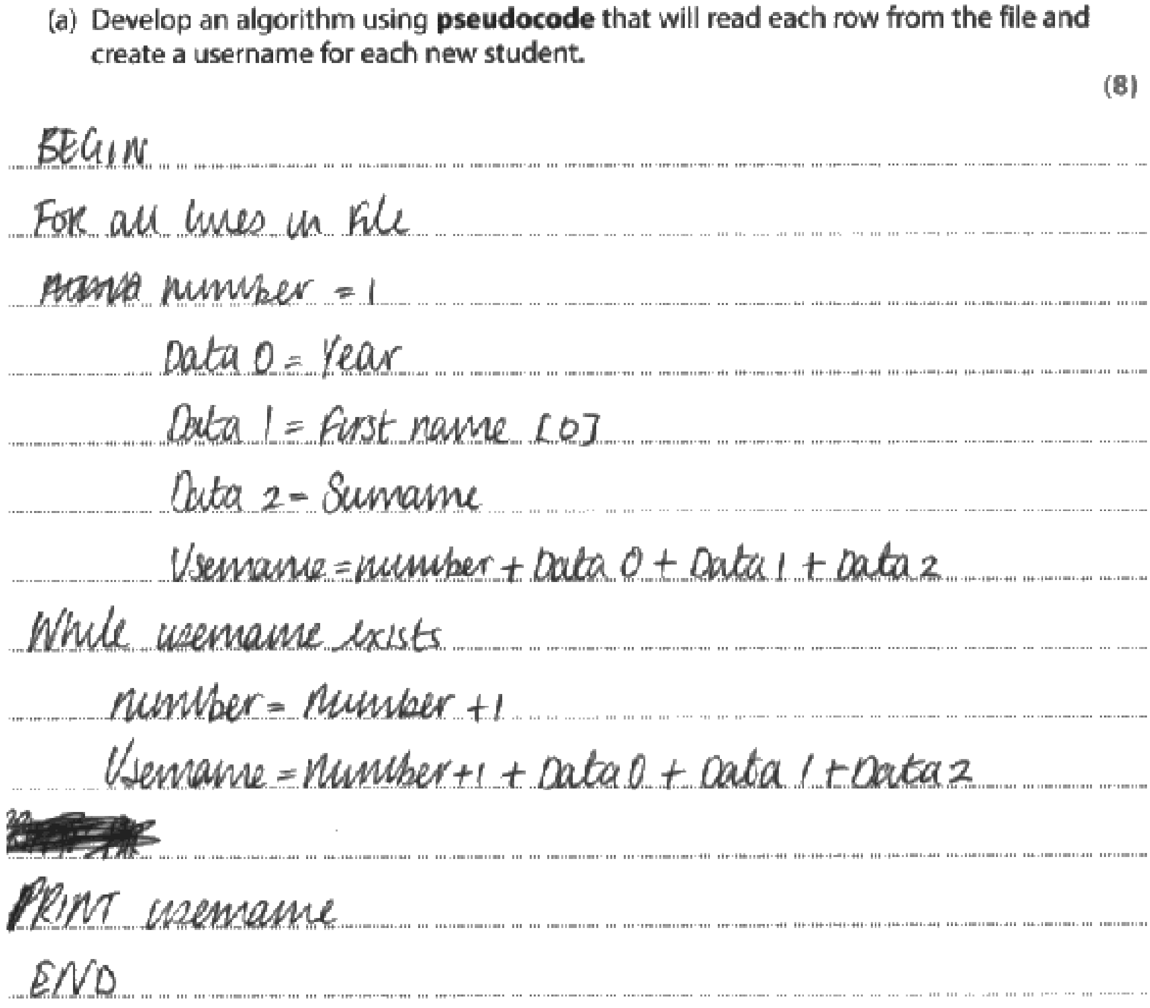
Structure – The learner has structured their algorithm well. The learner has made correct use of hierarchies/subdivision by using indentation to show code when they have used IF…ELSE statements and to show code repeating in the repeat until…loop.

Names – The learner has not used sensible variable names (e.g. Part A, Part B, Part C) however, these are all used consistently throughout the solution.

Logical operations – The learner has made good use of logic to actually put the username together but concatenating the different parts together. The learner has made use of IF….ELSE to search a username against those already created and used correct logic to increase the number. However when a new username is created there is no logic to then check this again.

Conventions - The conventions uses are generally well-known and used consistently. They have made use of begin, end, open, repeat, if, else etc.

Overall – The learner has provided an almost full and efficient solution. It is not complete because they have not used logic to keep checking the username until its unique. It currently will only check once.



(a) Develop an algorithm using **pseudocode** that will read each row from the file and create a username for each new student.

(8)

```
BEGIN
For all lines in File
~~name~~ number = 1
        Data 0 = Year
        Data 1 = First name [0]
        Data 2 = Surname
        Username = number + Data 0 + Data 1 + Data 2
While username exists
      number = Number + 1
      Username = Number+1 + Data 0 + Data 1 + Data 2
PRINT username
END
```

This solution meets the criteria for band 3 and 7 marks given.

Structure – The learner has structured their algorithm well. The learner has made correct use of hierarchies/subdivision by using indentation to show code being repeated for the *for* and *while* loop.

Names – Data0, Data1, Data2 are not appropriate but they are used consistently throughout the whole solution.

Logical operations – The learner has made good use of logic to actually put the username together using concatenation. The FOR loop would repeat for each line in the file, the WHILE loop would work until the username is unique. They have used correct logic to take the first letter of the first name and correct use of a number which increases by 1 if the username already exists.

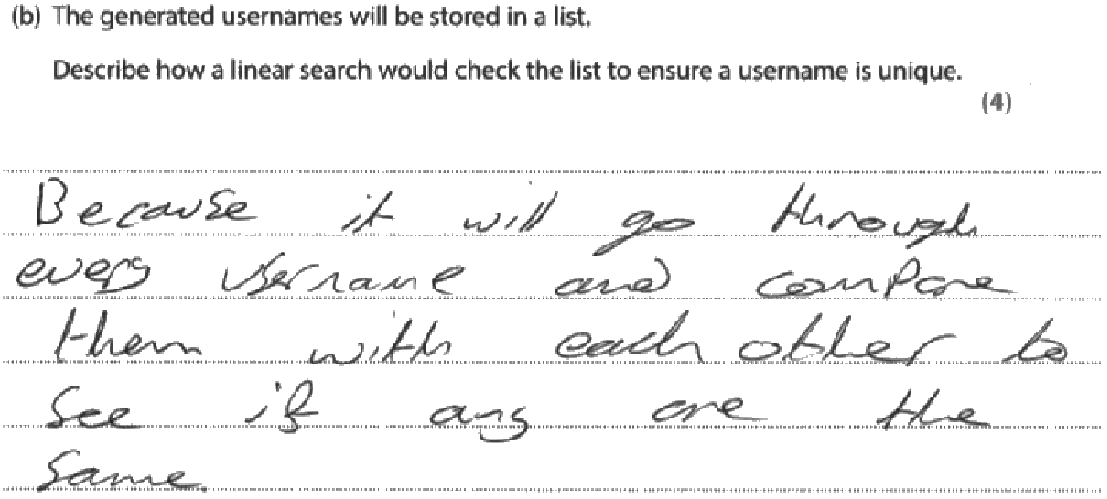
Conventions – The conventions uses are generally well-known and used consistently. They have made use of begin, end, while, for etc.

Overall – The learner has provided a full and efficient solution. The solution is short but actually addresses all points in the problem.

Question 3(b)

This question was poorly answered by most learners. Many learners did not know what a linear search was which therefore prevented them from gaining any marks. Many learners incorrectly focused their answers on the linear search ensuring the username is unique by increasing the number at the end of the username by 1 rather than the linear search searching the newly generated username against a list of existing ones. Some learners could state the linear search will search all items but very few learners understood how a linear search actually works.

This question is an 'Explain' style question for 4 marks which suggests that one point needed to be stated with a further three linked expansions. This question therefore demanded more depth from the learners. Some learners managed to state a point an initial point and every few manage to expand it. Hardly any learners were able to show more depth and get more than 2 marks for this question.



(b) The generated usernames will be stored in a list.

Describe how a linear search would check the list to ensure a username is unique.

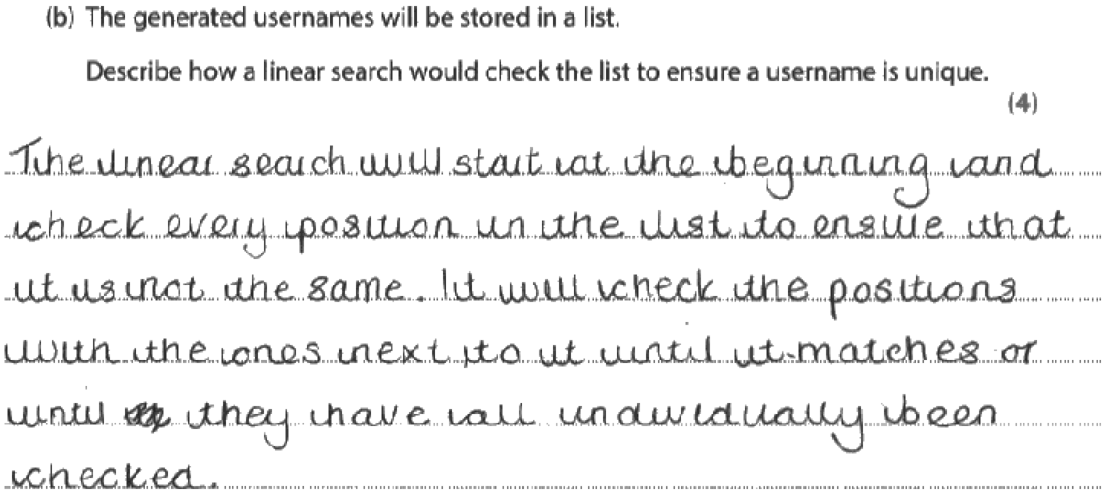
(4)

Because it will go through every username and compare them with each other to See if any are the Same

1 mark given for:

'Because it will go through every username'(1) there is enough evidence here to suggest that all usernames are checked.

'and compare them with each other to see if any are the same' - there is not enough evidence here to achieve the comparison mark as it does not show an understanding of comparing a new username to those already created (search criteria).



(b) The generated usernames will be stored in a list.

Describe how a linear search would check the list to ensure a username is unique.

(4)

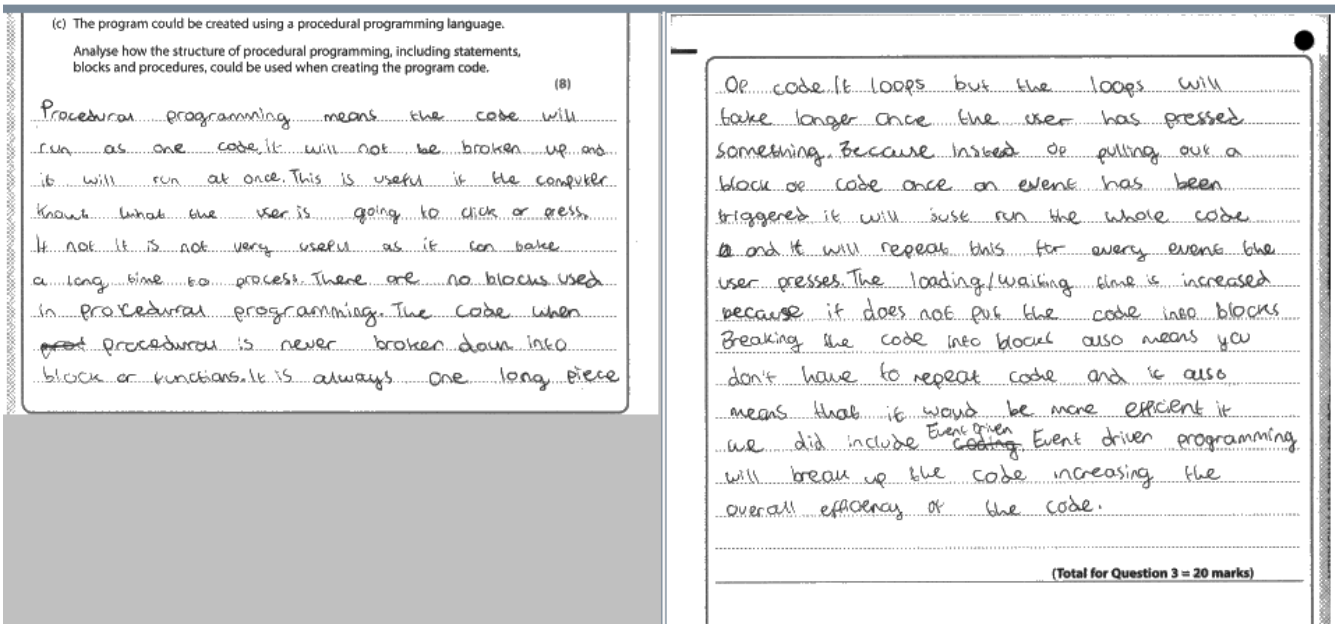
The linear search will start at the beginning and check every position in the list to ensure that it is not the same. It will check the positions with the ones next to it until it matches or until they have all individually been checked.

3 marks given for:

‘The linear search will start at the beginning’ (1)
‘will check every position in the list’ (1)
‘It will check the positions with the ones next to it' - is not accurate and cannot be given a mark
'until it matches' (1)
'until they have all individually been checked’ – There is enough evidence in that last point to show an understanding that the search process is repeated for each position. However, this mark point has already been achieved.

Question 3(c)

This question was very poorly answered by most learners. Unfortunately almost all learners did not know what a statement, block or procedure was despite these being stated in the specification which therefore meant that most learners achieved 0 marks. Some learners managed to talk about other features of procedural programming which were given credit.
Learners were required to analyse these feature by stating what they are and how the programming could make use of these when creating the programming code for the program in the given scenario. As learners did not have the subject knowledge for this question, they were unable to do this.



(c) The program could be created using a procedural programming language.

Analyse how the structure of procedural programming, including statements, blocks and procedures, could be used when creating the program code.

(8)

Procedural programming means the code will run as one code. It will not be broken up and it will run at once. This is useful if the computer knows what the user is going to click or press. If not it is not very useful as it can take a long time to process. There are no blocks used in procedural programming. The code when ~~proc~~ procedural is never broken down into block or functions. It is always one long piece of code. It loops but the loops will take longer once the user has pressed something. Because instead of pulling out a block of code once an event has been triggered it will just run the whole code ~~a~~ and it will repeat this for every event the user presses. The loading/waiting time is increased because it does not put the code into blocks. Breaking the code into blocks also means you don't have to repeat code and it also means that it woud be more efficient if we did include Event driven ~~coding~~. Event driven programming will break up the code increasing the overall efficency of the code.

(Total for Question 3 = 20 marks)

This response meets the criteria for band 1 and 1 mark given.

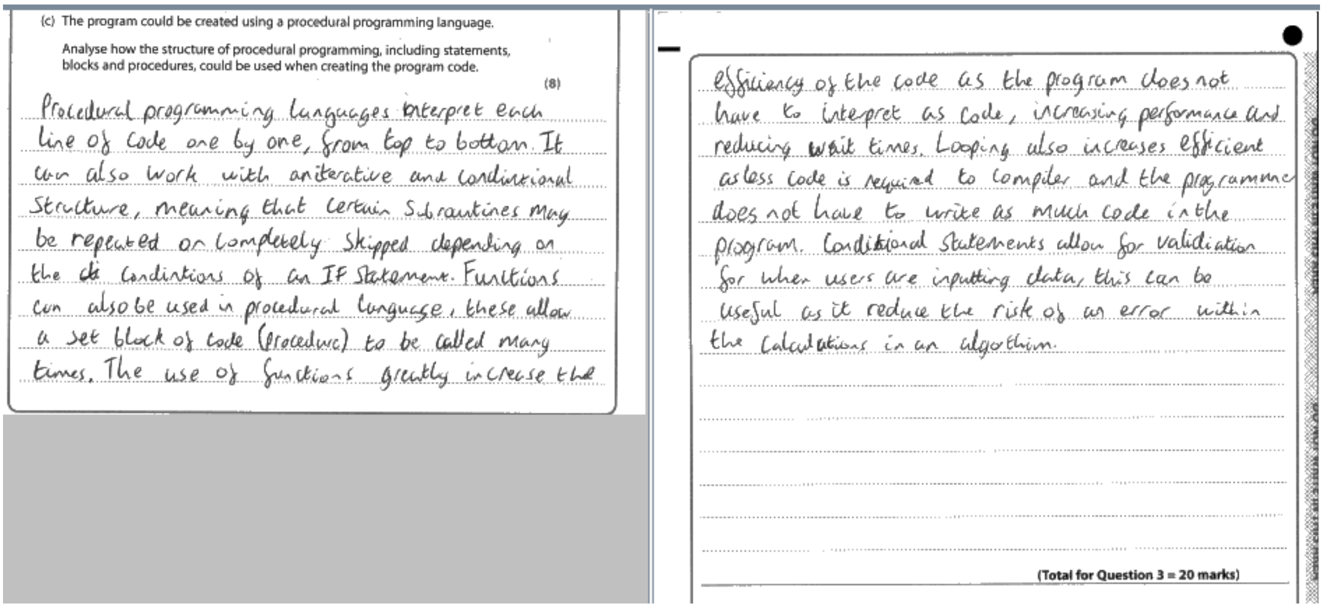
Technical vocabulary/Arguments – The response contains various inaccurate arguments such as "code is not broken up" "there are no blocks used" "it's always one long piece of code"

Application to question – "breaking the code into blocks means you don't have to repeat code." is accurate but no attempt is made at all to relate procedural programming to the scenario

Balance – Although the learner has included the main key words from the question, the learner has now provided any analysis of how they could be used.

Chains of reasoning – The learner has attempted to give chains of reasoning how these are always inaccurate.

Overall – This is a very poor responses. It contains inaccurate arguments throughout and points and backed up with further inaccurate arguments. There is not attempt at all to relate procedural programming to the question.



(c) The program could be created using a procedural programming language.

Analyse how the structure of procedural programming, including statements, blocks and procedures, could be used when creating the program code.

(8)

Procedural programming languages interpret each line of code one by one, from top to bottom. It can also work with aniterative and conditional structure, meaning that certain subrautines may be repeated or completely skipped depending on the conditions of an IF Statement. Functions can alsobe used in procedural language, these allow a set block of code (procedure) to be called many times. The use of functions greatly increase the efficiency of the code as the program does not have to interpret as code, increasing performance and reducing wait times. Looping also increases efficient asless code is required to compiler and the programmer does not have to write as much code in the program. Conditional statements allow for validiation for when users are inputting data, this can be useful as it reduce the risk of an error within the calculations in an algothim.

(Total for Question 3 = 20 marks)

This response meets the criteria for band 2 and 4 marks given.

Technical vocabulary/Arguments - "Procedural programming interpret each line of code one by one" "it can work with an iterative and conditional structure meaning that certain sub routines may be repeated" "functions…these allow a set block of code to be called many times" however these points are very generic and not specific to the question.

Application to question – No attempt is made at all to relate procedural programming to the scenario

Balance – The learner has covered functions and sub-routines and managed to give a brief expansion although not relevant to the scenario.

Chains of reasoning – One good chain of reasoning is used "Functions can be used" and then followed up with "these allow a set block of code to be called" and then followed up with "the functions increase the efficiency" and then followed up with "increasing performance"

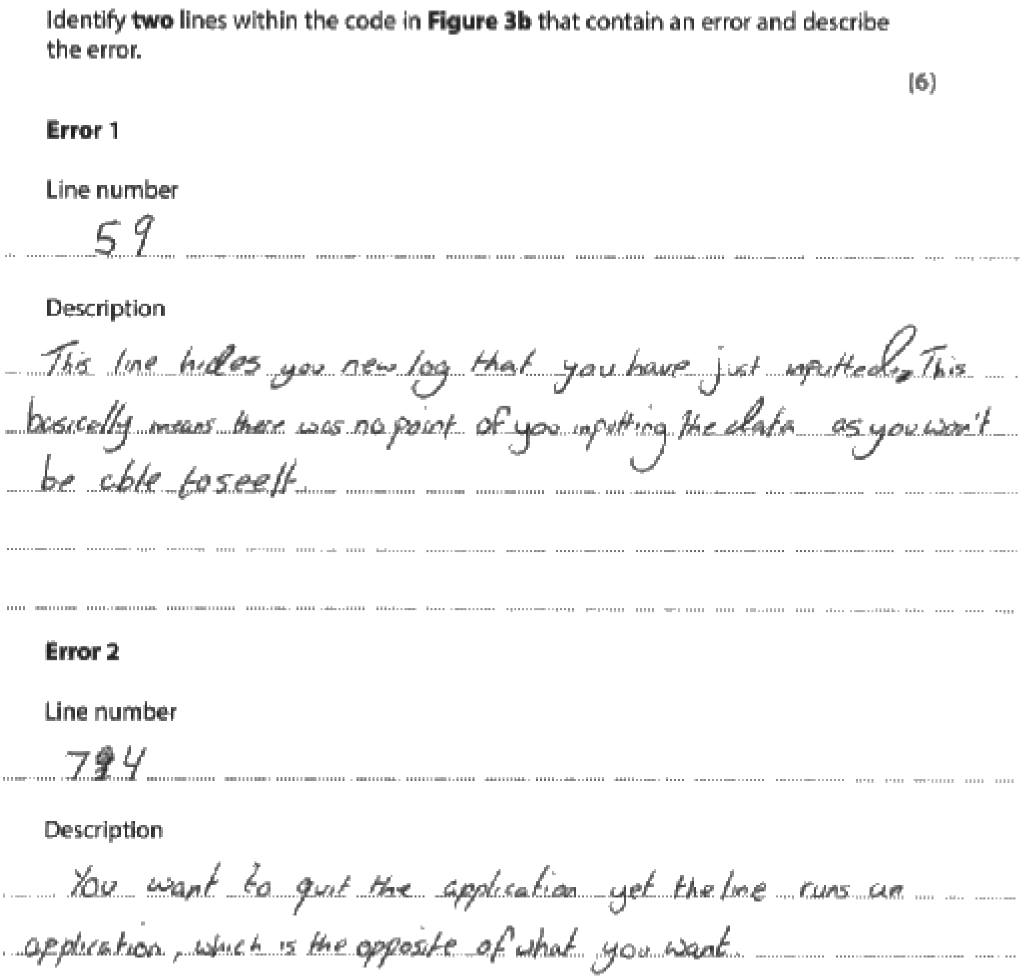
Overall – The learner has raised some good points in places and although they are relevant to procedural programming, they are not always relevant to the scenario. The learner has attempted to give chains of reasoning.

Question 4(a)

This question was very well answered with lots of learners scoring full marks. Most learners managed to identify the two lines of the code that contained an error and most learners were able to expand this with valid reasons.

It is worth noting that this programming code was written in C# as stated in the specification. Learners may also be given programming code written in Python 3.4 , Visual Basic and HTML 5.

This question is an ‘Identify and Describe’ style question for 6 marks which suggests that two points should have been stated for two lines of code that contained the error with a further expansions of two points for a description for each error for a further 4 marks.



Identify **two** lines within the code in **Figure 3b** that contain an error and describe the error.

(6)

Error 1

Line number

59

Description

This line hides you new log that you have just inputted. This basically means there was no point of you inputting the data as you won't be able to see it.

Error 2

Line number

74

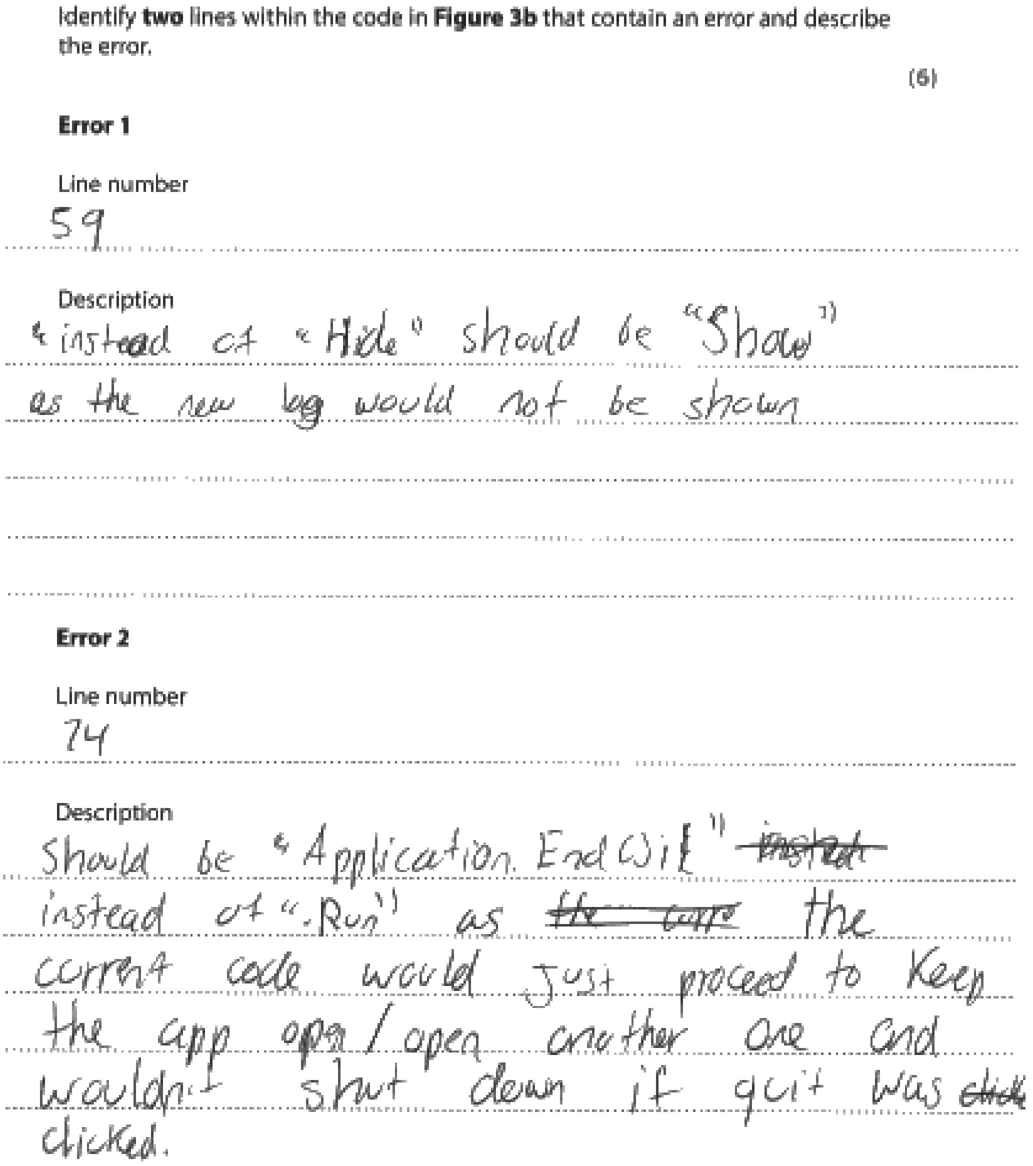
Description

You want to quit the application yet the line runs an application, which is the opposite of what you want.

4 marks given for:

Line ‘59’ (1) ‘The line hides you new log’ (1) ‘ – Not enough evidence here to suggest when this happens (eg when the log button is pressed)

Line ‘74’ (1) ‘You want to quit the application yet the line runs an application’ (1) ‘ – Not enough evidence here to suggest when this happens (eg when the quit button is pressed)



Identify **two** lines within the code in **Figure 3b** that contain an error and describe the error.

(6)

Error 1

Line number

59

Description

“instead of “Hide” should be “Show”
as the new log would not be shown

Error 2

Line number

74

Description

Should be “Application.EndWit” ~~instead~~
instead of “.Run” as ~~the code~~ the
current code would just proceed to keep
the app open / open another one and
wouldn't shut down if quit was ~~click~~
clicked.

6 marks given for:

Line ‘59’ (1) ‘instead of hide should be show (1) as the new log would not be shown’ (1) - this description is just enough for 2 marks

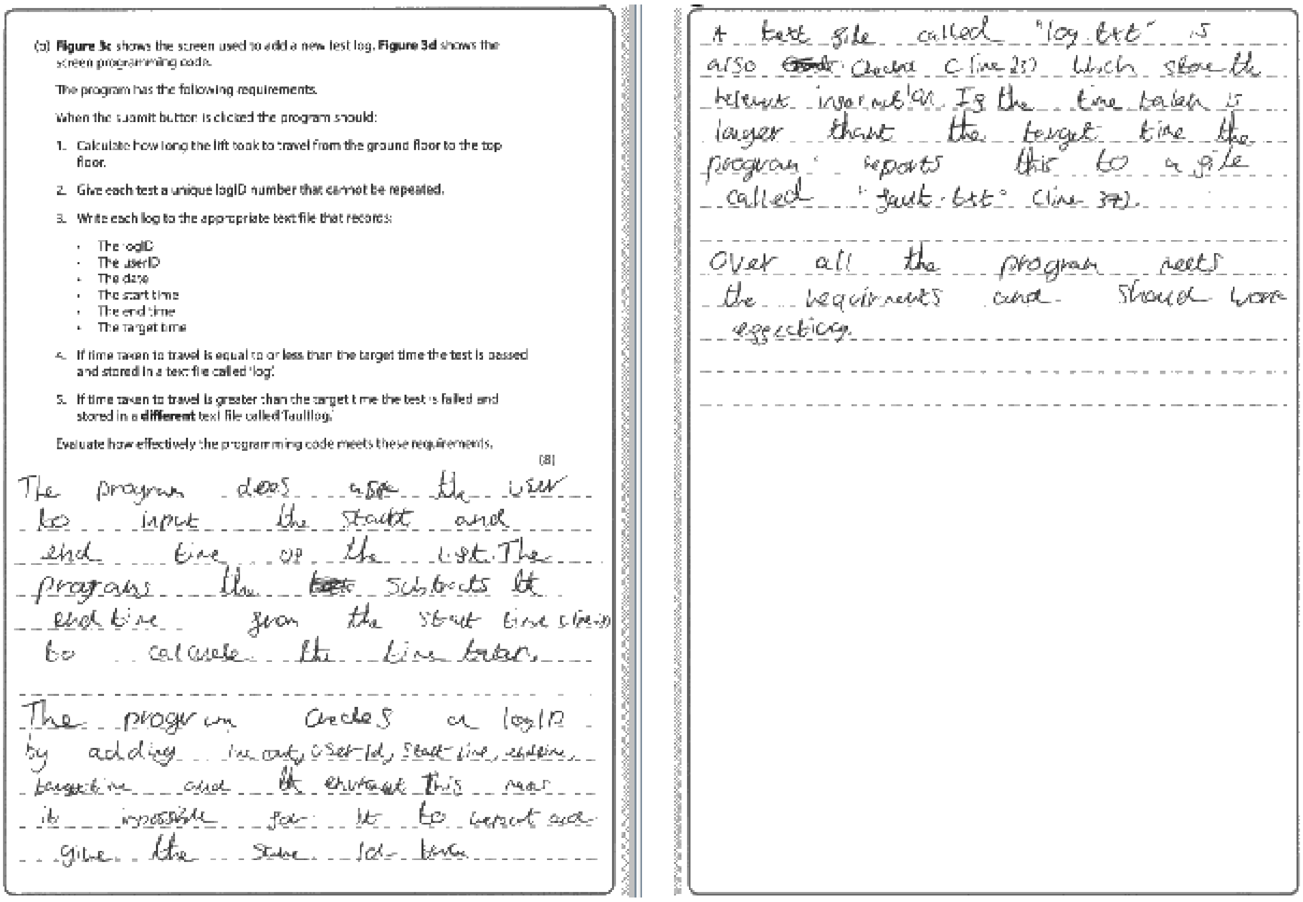
Line ‘74’ (1) ‘Should be application.end instead of .run’ (1) ‘it wouldn’t shut down if the quit was clicked’ (1)

Question 4(b)

This question was poorly answered by most learners. The vast majority of learners were not able to interpret C# code. It is worth noting that learners are expected to analyse, interpret and debug programming code written in Python 3.4, C Family, Visual Basic and HTML 5.

This question was an 'evaluate' style question were learners had to evaluate how effectively the programming code meets the given requirements. Learners should have taken some of the programming code and then considered the positive and negative ways that the code meets the requirements. As this is an evaluation question, learners were also expected to give a conclusion. For mark bands 2 and 3 the conclusion should make summary reference to the points considered throughout the response in order to make a supported judgement. 'e.g. This would be beneficial because…' As the vast majority of learners were not able to interpret the programming code, many learners scores 0 marks.

Many learners simply went through each requirement and simply guessed if the requirement had been met or not which did not show enough understanding for credit to be given. Learners that picked up some marks were able to state that the requirement was met / not met and then link this to the code and backup their points with evidence. 'e.g. requirement 1 has been met because on line ……… this clearly shows…….'



(a) **Figure 3c** shows the screen used to add a new test log. **Figure 3d** shows the screen programming code.

The program has the following requirements.

When the submit button is clicked the program should:

1. Calculate how long the lift took to travel from the ground floor to the top floor.
2. Give each test a unique logID number that cannot be repeated.
3. Write each log to the appropriate text file that records:
   - The logID
   - The userID
   - The date
   - The start time
   - The end time
   - The target time
4. If time taken to travel is equal to or less than the target time the test is passed and stored in a text file called 'log'.
5. If time taken to travel is greater than the target time the test is failed and stored in a **different** text file called 'faultlog.'

Evaluate how effectively the programming code meets these requirements.

[8]

The program does ask the user to input the start and end time of the list. The programs then ~~[illegible]~~ subtracts the end time from the start time (line [illegible]) to calculate the time taken.

The program creates a logID by adding the date, user-Id, start time, end time, target time and the encrypts this means it is impossible for it to repeat and give the same Id twice.

A text file called "log.txt" is also ~~[illegible]~~ created (line 23) which stores the relevant information. If the time taken is larger than the target time the program reports this to a file called "fault.txt" (line 37).

Over all the program meets the requirements and should work effectively.

This response meets the criteria for band 2 and 3 marks given.

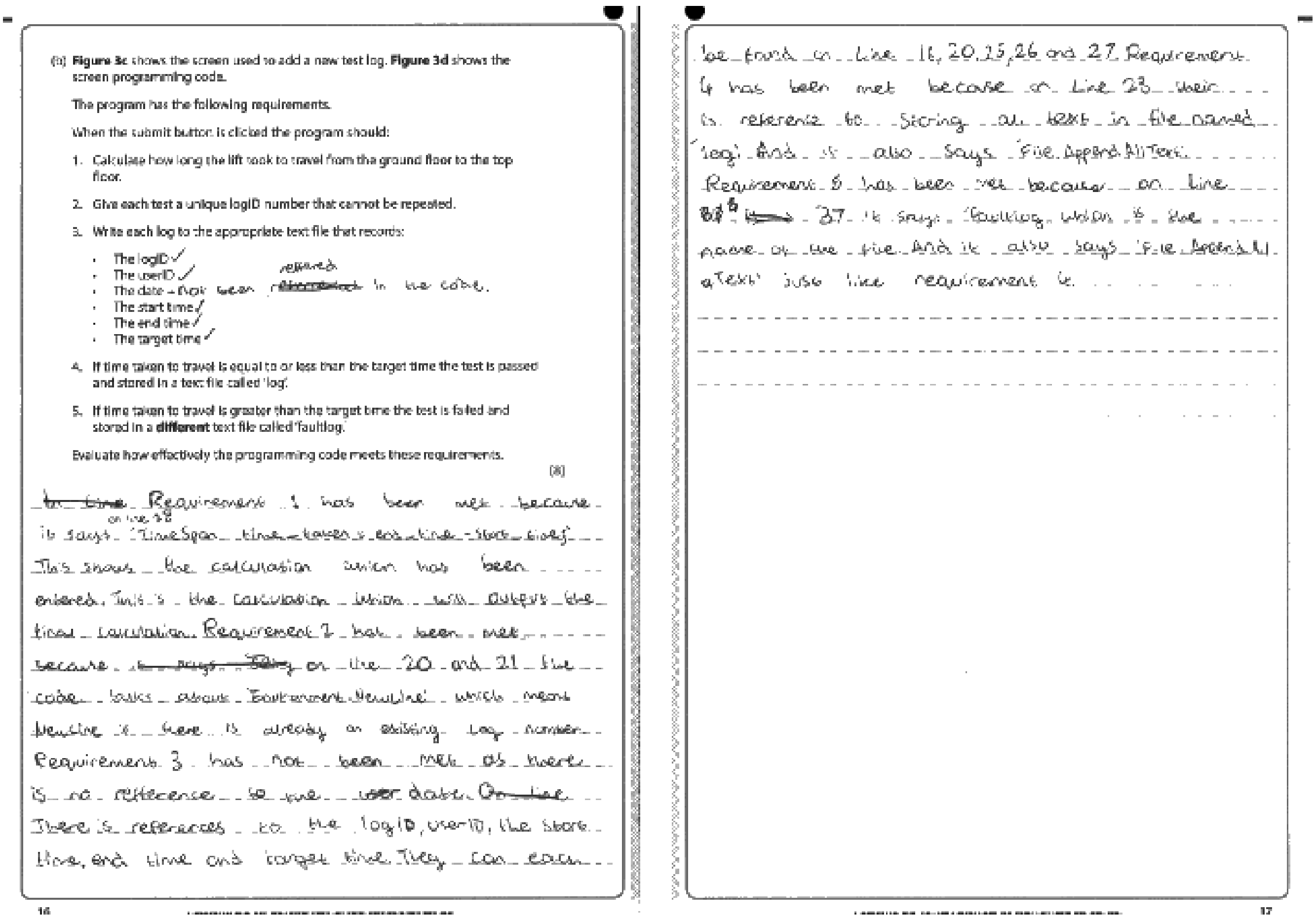
Technical vocabulary/arguments – The learner has mostly made accurate points throughout such as "log.txt which stores relevant information" "if the time taken is longer than the target time the program reports this to a file called fault.txt.

Balance – The learner has attempted to cover a range of different requirements and has correctly identified when some has been met. However the amount of information given is limited.

Chains of reasoning – Overall the learner has correctly where some of the requirements have been met and has provided a brief description and used evidence (from the code) to back up their answers.

Conclusion - Conclusion is not sufficient.

Overall – Overall the learner has correctly identified where some of the requirements being met and has provided a brief description and used evidence from the code to back up their answers. No conclusion has been met and there is still a large part of the response that is inaccurate.



(h) **Figure 3c** shows the screen used to add a new test log. **Figure 3d** shows the screen programming code.

The program has the following requirements.

When the submit button is clicked the program should:

1. Calculate how long the lift took to travel from the ground floor to the top floor.
2. Give each test a unique logID number that cannot be repeated.
3. Write each log to the appropriate text file that records:
   - The logID ✓
   - The userID ✓
   - The date – have been ~~[illegible]~~ refered in the code.
   - The start time ✓
   - The end time ✓
   - The target time ✓
4. If time taken to travel is equal to or less than the target time the test is passed and stored in a text file called 'log'.
5. If time taken to travel is greater than the target time the test is failed and stored in a **different** text file called 'faultlog'.

Evaluate how effectively the programming code meets these requirements.

(8)

~~to time~~ Requirement 1 has been met because on line 28 it says 'TimeSpan time taken = end time - start time' This shows the calculation which has been entered. This is the calculation which will output the time calculation. Requirement 2 has been met because ~~it says they~~ on line 20 and 21 the code talks about 'Environment.NewLine' which means NewLine if there is already an existing log number. Requirement 3 has not been met as there is no reference to the ~~user~~ date. ~~On line~~ There is references to the logID, user-ID, the start time, end time and target time. They can each be found on line 16, 20, 25, 26 and 27. Requirement 4 has been met because on line 23 their is reference to storing all text in file named 'log'. And it also says 'File.AppendAllText'. Requirement 5 has been met because on line ~~8~~ 27 it says 'faultlog' which is the name of the file. And it also says 'File.AppendAllText' just like requirement 4.

This response meets the criteria for band 2 and 4 marks given.

Technical vocabulary/arguments – The learner has correct identified that requirement 1 is met on line 28, the information provided for requirement 2 is inaccurate, the learner has correctly identified that requirement 3 is not met because the data is not referred to, the information provided for requirement 4 and 5 are inaccurate.

Balance – The learner has attempted to cover all five requirements and has correctly identified that two are not met.

Chains of reasoning –Some chains of reasoning are used in placed. The learner has identified that requirement 1 has been met and requirement 3 has not been met. They have then used chains of reasoning to state the line numbers / write out small parts of the code to back up their points.

Conclusion – No conclusion is given.

Overall – Overall the learner has correctly identifying 2 requirements being met/not met and has provided a brief description and used evidence to back up their answers. No conclusion has been met and there is still a large part of the response that is inaccurate.

Question 4(c)

This question was answered well by some learners. This was a very open question that allowed learners to discuss their knowledge both from this unit, other units and from their general experiences of programming. It is worth noting that unit 1 is a synoptic unit and learners should be encouraged to link as much as they can to other units within the qualification.
There were a lot of learners that left this question empty or who had a good attempt but scored 0 marks. Learners that did score marks generally tended to focus their answers on errors being generated when translating the code, the differences in built-in functions and the amount of support available for each language. However the vast majority of learners were only able to give short statements about each therefore resulting in many learners achieving marks within the first mark band.
The question asked learners to discuss the implications on developers but very few learners managed to do this. In order to get into mark band 2 or band 3, learners needed to state an implication and then describe in detail how this would affect the programmer. E.g. ‘If a programmer is switching to a different language that does not have the same built-in functions, this would increase the programmers workload. This is because the programmer will be required to create the functions themselves by writing out the code. This code will also need to be tested which increases workload further. If the programmer is not fully familiar with the programming language then they may not be able to debug the errors….’

(c) Richard would like to add extra features to the program, however the original programmer is not available.

The new programmer is not familiar with the programming language that has been used and intends to translate it into a different language.

Discuss the implications for developers of translating code into another programming language.

(12)

Different programming languages are designed for different tasks and different hardware in some cases. Not all languages have same functions so program may not be completely translatable. Translation software can often make mistakes during translation meaning a programmer that understands both languages needs to inspect old and new code for consistency to ensure new program functions as originally designed. It is possible that the code may need to be modified again beyond this point and the original creator might not know the new language. Different languages manage UI differently so interface may look different requiring time to get used to new UI. New code may not perform optimally as the translated version may work but not be the correct way to do the task.

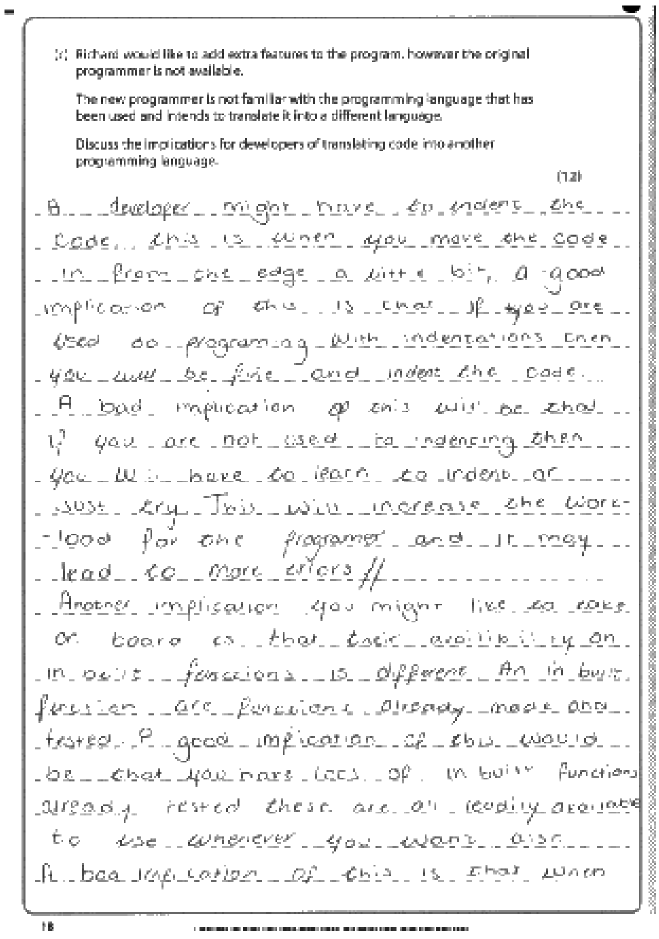
Technical vocabulary/arguments – The learner has raised some very good implications for programmers. They have mentioned specific points but the depth is limited.

This response meets the criteria for band 1 and 4 marks given.

Balance – The learner has attempted to cover different points such as functions, errors, knowledge, user interfaces however their depth of these is limited.

Chains of reasoning – The learner has raised some points such as "not all programming languages have same functions" although backed up with a weak point. Other points are not expanded such as "so program may not be completely translatable," "a programmer will need knowledge of both programming languages," "different languages make UI differently so interface may look different."

Overall –Although chains of reasoning is not given, the leaner has provided some very good, although brief, implications. It is however, not clear why the points they raised are important or how they actually impact on the programmer when translating code. There is just enough for the top of mark band 1 to be achieved.



(c) Richard would like to add extra features to the program, however the original programmer is not available.

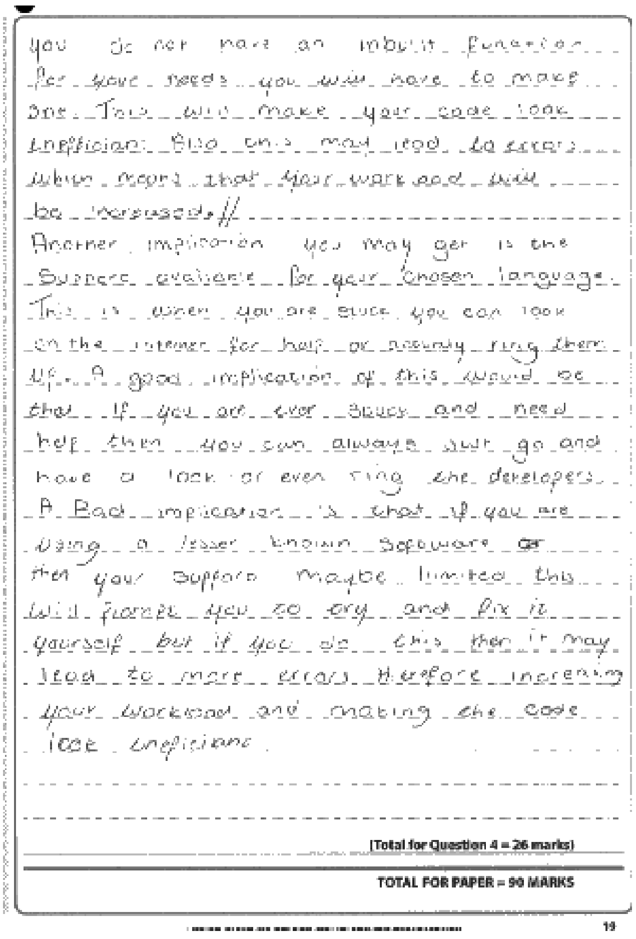
The new programmer is not familiar with the programming language that has been used and intends to translate it into a different language.

Discuss the implications for developers of translating code into another programming language.

(12)

A developer might have to indent the code, this is when you move the code in from the edge a little bit, a good implication of this is that if you are used to programing with indentations then you will be fine and indent the code. A bad implication of this will be that if you are not used to indenting then you will have to learn to indent or just try. This will increase the work-load for the programer and it may lead to more errors //

Another implication you might like to take on board is that their avilibility on in built functions is different. An in built fuction are functions already made and tested. A good implication of this would be that you have lots of in built functions already tested these are all readily avaiable to use whenever you want. also A bad implication of this is that when



you do not have an inbuilt function for your needs you will have to make one. This will make your code look inefficiant. Also this may lead to errors which means that your workload will be increased. //

Another implication you may get is the support available for your chosen language. This is when you are stuck you can look on the internet for help or actually ring them up. A good implication of this would be that if you are ever stuck and need help then you can always just go and have a look or even ring the developers. A Bad implication is that if you are using a lesser known software ~~or~~ then your support maybe limited this will force you to try and fix it yourself but if you do this then it may lead to more errors therefore increasing your workload and making the code look inefficiant.

(Total for Question 4 = 26 marks)

TOTAL FOR PAPER = 90 MARKS

This response meets the criteria for band 2 and 8 marks given.

Technical vocabulary/arguments – Technical language is used accurately in most places. The learner backs up their points with their own knowledge. They use their own knowledge throughout to support their arguments.

Balance – The learner has covered three different areas - built-in functions, indentation, support and the learner has provided a very detailed description of each.

Chains of reasoning – Lots of chains of reasoning are used throughout. For example:

"availability of built-in functions.....they are already made and tested...if they are not available you have to make your own....which will make the code look inefficient....may lead to errors....workload will be increased.

Overall –The learner has raised carefully considered points that are relevant to 'translating code.' They have used chains of reasoning throughout and use their knowledge to accurately backup their points in most places. The learner has covered a range of issues and has clearly stated the good and bad implication of each. However the first section about indentation is not relevant.

Summary

Based on performance in this examination series, learners are offered the following advice:

- Develop more of an understanding of logic. It is recommended that learners are used to developing algorithms both as flowcharts and Pseudocode. This will also allow learners develop their understanding of a lot of the different areas on the specification in an applied way. It may be best for learners to do this alongside the unit which increases in complexity over time rather than being taught in one go. Being able to create effective algorithms will help in all learning aims in this unit.
- Develop understanding of key terminology used in the unit so that you are able to access the context of the question.
- Apply their knowledge to as many different scenarios as possible to prepare learners to be able to provide answers/information to the given context under exam conditions.
- Develop understating of the requirements of the different command verbs used in the unit so that you can structure your response appropriately in order to maximise the marks you achieve. A lot of the different command verbs and their meaning can be found in the specification.
- For shorter response questions (5 marks or less), make note of the number of marks available this will help you identify the number of points you need to make. For example, a 4 mark 'Explain one…' style question would need to make at least four linked points that expand/exemplify understating of a single point
- When producing extended writing responses (6 marks or more) ensure you consider a range of points, each of which should be expanded or supported with examples and applied to the given context.

Pearson Education Limited. Registered company number 872828
with its registered office at 80 Strand, London, WC2R 0RL, United Kingdom