Examiners' Report
Lead Examiner Feedback

January 2022

Pearson BTEC L3 Nationals
In Computing (31771H)
Unit 4: Software design and development project

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 website at http://qualifications.pearson.com/en/home.html for our BTEC qualifications.

Alternatively, you can get in touch with us using the details on our contact us page at http://qualifications.pearson.com/en/contact-us.html

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:
http://qualifications.pearson.com/en/support/support-for-you/teachers.html

You can also use our online Ask the Expert service at https://www.edexcelonline.com
You will need an Edexcel Online 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

January 2022
Publications Code 31771H_2201_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, at Distinction, Merit and Pass.

Setting grade boundaries

When we set grade boundaries, we look at the performance of every learner who took the external 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 is 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 external assessment 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 assessment, because then it would not take accessibility into account.

Grade boundaries for this, and all other papers, are on the website via this link:
http://qualifications.pearson.com/en/support/support-topics/results-certification/grade-boundaries.html

Awarding BTEC qualifications in 2022

Ofqual has set out their plans for awarding qualifications in 2022 and intend to return to a normal, pre-pandemic, approach to grading standards over by 2023. They have confirmed that 2022 will be a transition year, to reflect that we are in a pandemic recovery period and students' education has been disrupted.

Our guiding principle and approach to awarding BTEC qualification results in 2022 will be to ensure parity in relation to the approach being taken for GCSE and A level learners. BTEC courses have a different structure and design to academic qualifications - BTECs are modular qualifications (with assessments taking place throughout the course) compared to GCSEs and A levels which are linear (assessed and awarded at the same time at the end of the year), and therefore our approach needs to be different.

In 2022 we will return to the usual method of calculating BTEC qualification results, however adaptations including, U-TAGs and reduced internal assessment, are in place to provide a comprehensive package of support for students.

The basis of our awarding approach to BTECs this year is to ensure it is as fair as possible for all learners. We will use a range of evidence to set grade boundaries for the external units. Part of this evidence will be to closely monitor learner performance in all assessments that contribute to learners' final qualification grade, to ensure parity with A level and GCSEs.

Further information can be found on our website and via our Social Media channels.

31771H: Software design and development project

Grade	Unclassified	Level 3			
		N	P	M	D
Boundary Mark	0	10	21	35	50

Introduction

This was the sixth examination series for Level 3 BTEC Computing Unit 4: Software design and development project.

This unit is a paper-based exam, assessed through a task-based assessment. The set task assesses learners' ability to design, create and evaluate software using Python (3.4 or a later version) or one of the C family programming languages. This unit is a mandatory unit for all learners studying the extended diploma.

The examination for this unit will always contain five activities and each one will be linked to a scenario. The scenario is clearly stated at the beginning of each assessment. The activities will test learners on different areas of the specification, and learners are expected to apply their knowledge to the scenario.

All Activities of the examination paper provide differentiation at all attainment levels and the brief is designed to escalate in difficulty so that a larger percentage of higher-grade marks depends on the skills, knowledge, understanding and application of theory.

Introduction to the Overall Performance of the Unit

The overall performance of learners was slightly better than the previous series for this unit.

The performance on Activity 1 resulted in most learners picking up marks in band 2. Majority of the responses seen used BCS symbols, the better responses were able to break down the requirements into relevant parts. Learners provided evidence of links between component parts but evidence of handling errors within the flowcharts was not always present. Many learners tried to fit the entire flowchart on to one page, which tended to make them difficult to follow. Many learners applied some validation, mostly to the lawn dimensions rather than the customer details. Learners working at the higher end of the mark range applied some kind of validation to the customer details such as presence checks or type checks.

Activity 2 was of generally of a good standard and demonstrated the learner ability to apply pseudocode design methodologies to the scenario. Some learners produced pseudocode that was very similar to the code produced

in activity 4. This reduces clarity and readability for anyone not familiar with the language specific functions or syntax. The aim of the pseudocode is to provide a step in the design process that would allow a third party if needed to continue with the coding and any of the specified languages.

Activity 3 & 4 (testing) was not well demonstrated which resulted in many learners only accessing band 1. It is recommended that centres reinforce what a test plan consists of and the importance of testing throughout the whole design and development process. In most cases, the testing carried out did not evidence any errors encountered which is essential for accessing higher marks. Learners mostly tested the inputs, with some limited data, stating simply that the data would be accepted / rejected. However most learners did not test the final calculations using known data and manually calculated results. This would then fully test the final bill. Calculating the VAT was one part of the calculation were mathematical logic errors (BODMAS) were made resulting in an incorrect final bill.

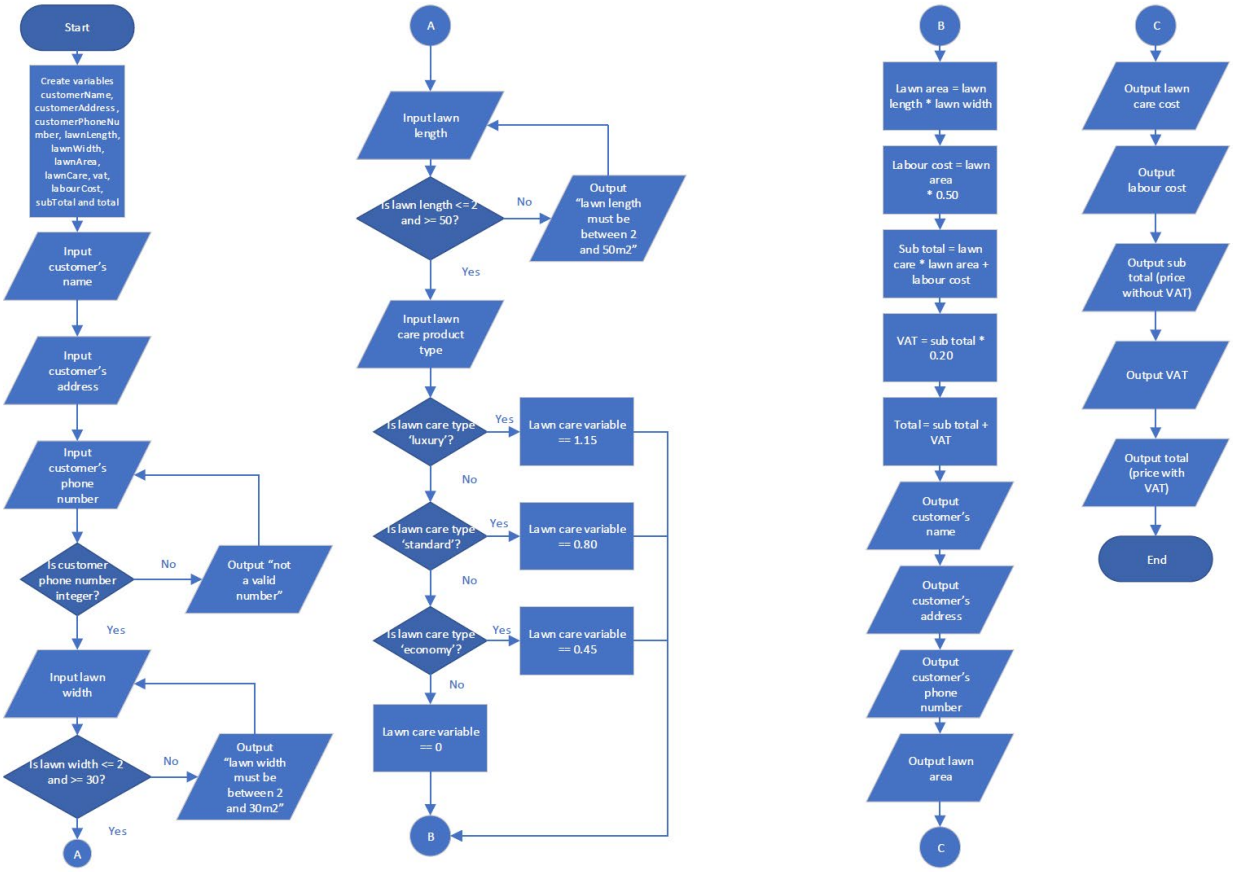
Activity 4 (Coding) was well completed by some learners with marks awarded in the top mark band as they produced a working solution along with detailed comments. These learners validated the inputs well and in trapped errors due to wrong data types being input by the user or the dimensions of the lawn being out of range. A common issue with validation was a lack of feedback to user indicating why the input had been rejected. Meaningful error messages would improve the code and the user experience.

The evaluations (activity 5) were of a good standard and most learner's accessed bands two and three. Some learners only produced a step-by-step account of what they did which resulted in marks from band 1 being awarded.

Individual Questions

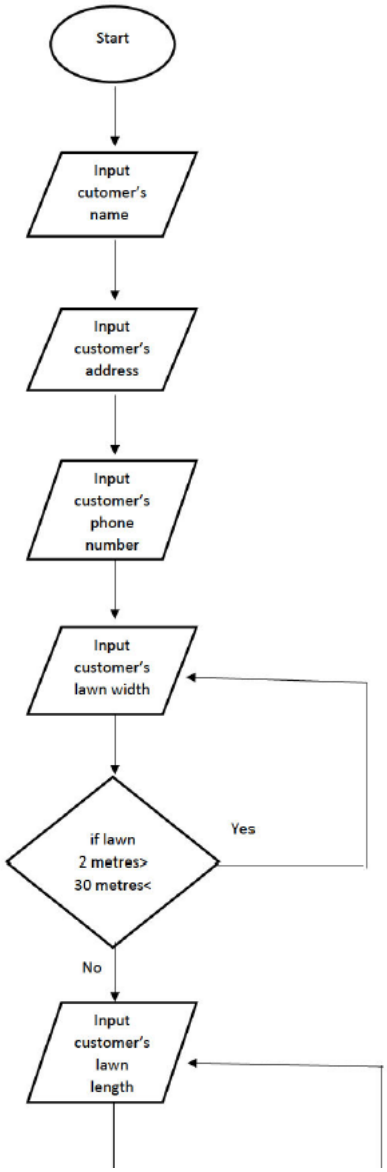
The following section considers each question on the paper, providing examples of learner responses and a brief commentary of why the responses gained the marks they did. This section should be considered with the live external assessment and the corresponding mark scheme.

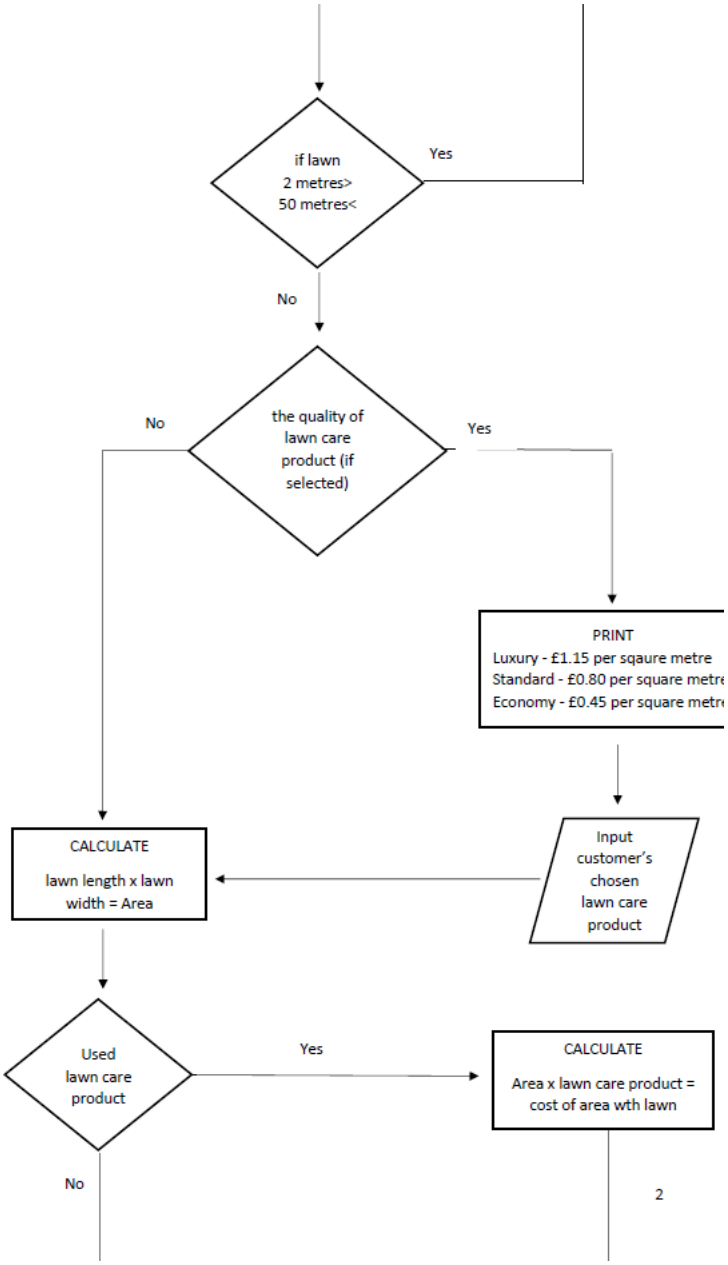
Activity 1

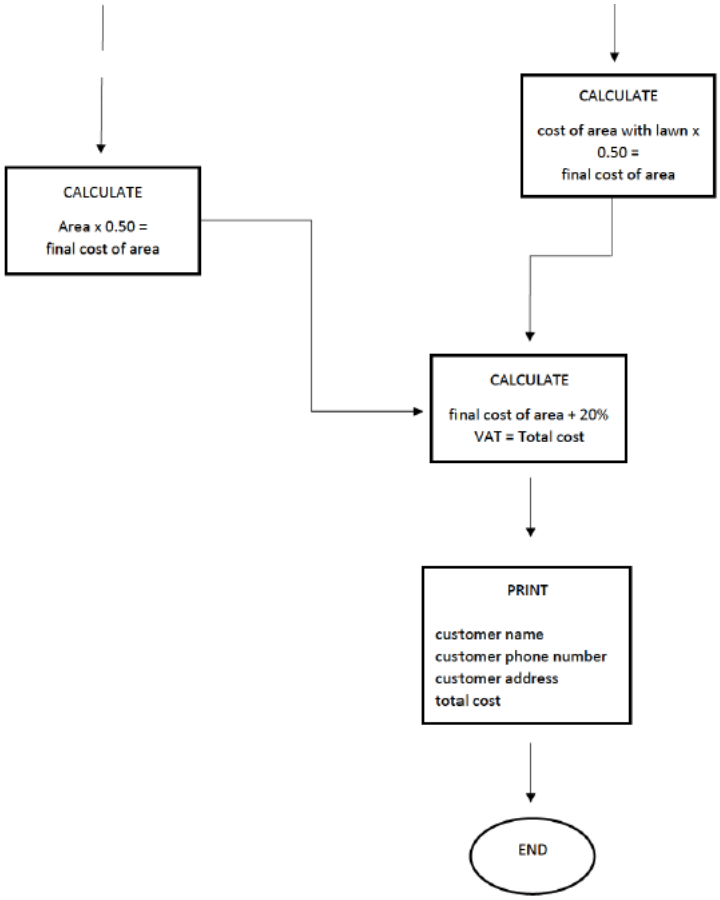


The learner shows accurate use of BCS symbols. The logic is correct, the flowchart is easy to follow and breaks down requirements into component parts. Shows full coverage of input and outputs, the naming conventions used are appropriate and consistent.

Band 3 (10 marks)


Start
Input cutomer's name
Input customer's address
Input customer's phone number
Input customer's lawn width
if lawn 2 metres> 30 metres<
Yes
No
Input customer's lawn length


if lawn
2 metres>
50 metres<
Yes
No
the quality of lawn care product (if selected)
No
Yes
PRINT
Luxury - £1.15 per sqaure metre
Standard - £0.80 per square metr
Economy - £0.45 per square metr
Input customer's chosen lawn care product
CALCULATE
lawn length x lawn width = Area
Used lawn care product
Yes
CALCULATE
Area x lawn care product = cost of area wth lawn
No
2



The learner shows use of BCS symbols, coverage of inputs, the leaner is inconsistent with naming conventions and uses of Y/N on decisions in flowchart therefore although the logic is difficult to follow. The logic for choosing and calculating the cost of lawn care product is also unclear.

Band 1 (3 marks)

Activity 2

```
BEGIN

DISPLAY Enter customers name
INPUT customersName

DISPLAY Enter customers address
INPUT customersAddress

DISPLAY Enter customers phone number
INPUT customersNumber

WHILE True DO
  DISPLAY Enter customers lawn width
  INPUT lawnWidth
  IF lawnWidth < 2 THEN
    DISPLAY Lawn width is to small
  ELSE IF lawnWidth > 30 THEN
    DISPLAY Lawn width is to big
  ELSE IF lawnWidth => 2 and lawnWidth =< 50 THEN
    EXITWHILE
  ELSE THEN
    DISPLAY Please enter a number between 2 and 50
  ENDIF
ENDWHILE

WHILE True DO
  DISPLAY Enter customers lawn length
  INPUT lawnLength
  IF lawnLength < 2 THEN
    DISPLAY Lawn length is to small
  ELSE IF lawnLength > 50 THEN
    DISPLAY Lawn length is to big
  ELSE IF lawnLength => 2 and lawnLength =< 50 THEN
    EXITWHILE
  ELSE THEN
    DISPLAY Please enter a number between 2 and 50
  ENDIF
ENDWHILE

lawnArea <- lawnLength * lawnWidth

WHILE True DO
  DISPLAY Does the customer want lawn care (y/n)
  INPUT addLawnCare
  IF addLawnCare = y THEN
    WHILE True DO:
      DISPLAY Enter quality of lawn care (e, s, l)
      INPUT lawnCare
      IF lawnCare = e THEN
        lawnCareFull = Economy
        careCost <- lawnArea * 0.45
```

```
            EXITWHILE
          ELSE IF lawnCare = s THEN
            lawnCareFull = Standard
            careCost <- lawnArea * 0.80
            EXITWHILE
          ELSE IF lawnCare = l THEN
            lawnCareFull = Luxury
            careCost <- lawnArea * 1.15
            EXITWHILE
          ELSE THEN
            DISPLAY Please enter e, s or l for Economy, Standard or Luxury
          ENDIF
        ENDWHILE
      ELSE IF addLawnCare = n THEN
        lawnCareFull <- None
        careCost <- 0
        EXITWHILE
      ELSE DO
        DISPLAY Please enter y or n
      ENDIF
    ENDWHILE


    labourCost <- lawnArea * 0.5

    totalCost <- labourCost + lawnCare

    totalCostVAT <- totalCost * 1.2



    DISPLAY
      Name: [customersName]
      Address: [customersAddress]
      Phone Numebr: [customersNumber]

      Lawn Width: [lawnWidth] M    Lawn Length: [lawnLength] M
      Lawn Area: [lawnArea] M2
      Lawn Care Type: [lawnCareFull]

      Labour Cost: [labourCost]
      Lawn Care Cost: [careCost]

      Total cost excluding VAT: £[totalCost]
      Total cost including VAT: £[totalCostVAT]
    ENDDISPLAY

    END
```

The learner has produced a structure which shows appropriate and consistent use of hierarchy and indentation, providing clarity and mostly readable pseudocode. The pseudocode will provide a working solution with some minor errors. Appropriate naming conventions have been used and precise use of logical operations.

Band 3 (8 marks)

START

FLOAT lawncare

FLOAT labour

FLOAT cost

FLOAT VAT

FLOAT VATcost

FLOAT length

FLOAT width

FLOAT totalarea

FLOAT squaremetre = 0.50

FLOAT lawncarecost

FLOAT m

STRING name

STRING address

STRING phonenumber

OUTPUT "Please replace any spaces which would be used with underscores when entering information"

OUTPUT "Please enter the name of the customer"

INPUT name

OUTPUT "Please enter the address of the customer"

INPUT address

OUTPUT "Please enter the phone number of the customer"

INPUT phonenumber

OUTPUT "Please enter the length of the lawn"

INPUT length

OUTPUT "Please enter the width of the lawn"

INPUT width

CALCULATION totalarea = length X width

CALCULATION labour = totalarea X squaremetre

OUTPUT "What quality of lawn care does the customer request? 1 standard 2 economy 3 luxury 4 none"

START

FLOAT lawncare

FLOAT labour

FLOAT cost

FLOAT VAT

FLOAT VATcost

FLOAT length

FLOAT width

FLOAT totalarea

FLOAT squaremetre = 0.50

FLOAT lawncarecost

FLOAT m

STRING name

STRING address

STRING phonenumber

OUTPUT "Please replace any spaces which would be used with underscores when entering information"

OUTPUT "Please enter the name of the customer"

INPUT name

OUTPUT "Please enter the address of the customer"

INPUT address

OUTPUT "Please enter the phone number of the customer"

INPUT phonenumber

OUTPUT "Please enter the length of the lawn"

INPUT length

OUTPUT "Please enter the width of the lawn"

INPUT width

CALCULATION totalarea = length X width

CALCULATION labour = totalarea X squaremetre

OUTPUT "What quality of lawn care does the customer request? 1 standard 2 economy 3 luxury 4 none"

OUTPUT "Please enter the number which the customer requires"

OUTPUT "Entering an invalid number will result in the program to restart"

INPUT m

IF (m == 1)

CALCULATION lawncare = 0.80

CALCULATION lawncare cost = lawncare X totalarea

CALCULATION cost = labour + lawncarecost

CALCULATION VAT = cost / 10 X 2

CALCULATION VATcost = cost + VAT

OUTPUT "The data which is below is given in the order of name, address, phone number, lawn care cost, labour cost, total cost before VAT, total cost after VAT and the total area"

OUTPUT name

OUTPUT address

OUTPUT phonenumber

OUTPUT lawncare

OUTPUT labour

OUTPUT cost

OUTPUT VATcost

OUTPUT totalarea

IF (m == 2)

CALCULATION lawncare = 0.45

CALCULATION lawncare cost = lawncare X totalarea

CALCULATION cost = labour + lawncarecost

CALCULATION VAT = cost / 10 X 2

CALCULATION VATcost = cost + VAT

OUTPUT "The data which is below is given in the order of name, address, phone number, lawn care cost, labour cost, total cost before VAT, total cost after VAT and the total area"

OUTPUT name

OUTPUT address

OUTPUT phonenumber

OUTPUT lawncare

```
        OUTPUT labour
        OUTPUT cost
        OUTPUT VATcost
        OUTPUT totalarea
IF (m == 3)
        CALCULATION lawncare = 1.15
        CALCULATION lawncare cost = lawncare X totalarea
      CALCULATION cost = labour + lawncarecost
        CALCULATION VAT = cost / 10 X 2
        CALCULATION VATcost = cost + VAT
 OUTPUT "The data which is below is given in the order of name, address, phone number, lawn
care cost, labour cost, total cost before VAT, total cost after VAT and the total area"
        OUTPUT name
        OUTPUT address
        OUTPUT phonenumber
        OUTPUT lawncare
        OUTPUT labour
        OUTPUT cost
        OUTPUT VATcost
        OUTPUT totalarea
IF (m == 4)
        CALCULATION cost = labour
        CALCULATION VAT = cost / 10 X 2
        CALCULATION VATcost = cost + VAT
        OUTPUT "The data which is below is given in the order of name, address, phone number,
labour cost, total cost before VAT, total cost after VAT and the total area"
        OUTPUT name
        OUTPUT address
        OUTPUT phonenumber
        OUTPUT labour
        OUTPUT cost



       OUTPUT VATcost
       OUTPUT totalarea
END
```

The learner has produced a structure that shows some appropriate hierarchies, readability is limited, and some sequences are repeated in linear fashion. The code uses appropriate naming conventions but there are inconsistencies. There is imprecise use of logical operations, leading to an incomplete solution.

Band 1 (3 marks)

Activity 3

Document for Activities 3 and 4

Test Plan (add additional rows as required)

Program language the product is to be produced in (tick box for language used):

Python ■ C Family □

Test Number	Purpose of test	Test Data	Expected Result	Actual Result	Comments
1	The phone number prompt should not accept non-integers	"a"	The program should say that it is not a valid phone number		
2	The phone number prompt should accept numbers	234788786	The program should accept it as a valid input and move on to next part		
3	The lawn width prompt must not accept strings or other characters	"a"	The program should say it is not valid		
4	The lawn width prompt must not accept numbers below 2 or higher than 30	0, 1, 31, 32	The program should say that the input should be between 3-30		
5	The lawn width prompt should accept numbers between 2 and 30	2, 5, 6, 19, 25, 29	The program should accept it and move onto lawn length		
6	The lawn length prompt must not accept strings or other characters	"a"	The program should say it is not valid and allow user to try again		
7	The lawn length prompt must not accept numbers below 2 or higher than 50	0, 1, 51, 52	The program should say that the input should be between 3-50		
8	The lawn length prompt should accept numbers between 2 and 50	2, 5, 6, 19, 25, 29, 34, 45, 49	The program should accept it and move onto lawn care prompts		

Test Number	Purpose of test	Test Data	Expected Result	Actual Result	Comments
9	Lawn care prompt must only accept "luxury", "standard", "economy" or "none"	"aaaaa"	The program should say that the input wasn't valid and allow the user to try again		
10	The lawn care prompt should accept one of "luxury", "standard", "economy" or "none"	"standard"	The program should accept it as a valid input and move on to next part		
11	Using a 30 × 25 m^2 lawn, the lawn area should be 750m^2.	Lawn width: "30" Lawn length: "25"	Lawn area should be 750sq m.		
12	Using a 30 × 50 m^2 lawn, the lawn area should be 1500m^2. (max size)	Lawn width: "30" Lawn length: "50"	Lawn area should be 1500sq m.		
13	Labour cost should be 50% of lawn area	Lawn width: "30" Lawn length: "50"	Labour cost should be £750.		
14	VAT should be 20% of subtotal.	Lawn width: "25" Lawn length: "10" Lawn care: "none"	VAT should work out to £25. Note that price includes labour cost		
15	Lawn care prompt should be case insensitive	"Standard"	The program should accept it as a valid input and proceed to move on		

16	The lawn length prompt should accept numbers between 2 and 50	2, 5, 6, 19, 25, 29, 34, 45, 49	The program should accept it and move onto lawn care prompts		
17	Customer details should be displayed correctly in results output.	Customer name: "alistair" Customer address: "test" Customer phone number: "1234"	All three of the test data results should be shown as they appear in the results output.		
18	Customer name should not accept numbers in input	Aaaa4	The program should not accept it and display a message to the user.		
19	Final output of prices needs to be formatted to 2 decimal places	Lawn width: 30 Lawn length: 43 Lawn care: none	Total should be		
20	The lawn width prompt must accept floats as numbers	Lawn width: 29.78	The program should say that the input should be between 3-30		
21	If no lawncare product selected, program should display a message saying that no lawncare product	Lawncare product type: "none"	The program should say that no lawncare product was selected instead of displaying the cost of		
22	Economy product rate should be 0.45	Lawn width: 30 Lawn length: 43 Lawn care: economy	Economy lawncare rate should be £0.45		
23	If customer phone number entered is 0, go to proceed to next prompt	Customer name: a Customer address: a Customer phone number: 0	Program should move on and to prompt for the lawn area width		

The learner has produced a thorough test plan to confirm a working solution which includes a range of data. Expected results are specific and accurate based on identified test data. The level of detail is not quite enough for the top of the mark band.

Band 3 (5 marks)

Document for Activities 3 and 4

Test Plan (add additional rows as required)

Program language the product is to be produced in (tick box for language used):

Python ☒ C Family ☐

Test Number	Purpose of test	Test Data	Expected Result	Actual Result	Comments
1	Testing the input "Customer name"	Peter Normal	The program should save the customer name and continue through the program		
2	Testing the input "Customer name"	Ye'a[rls232 Abnormal	The program should detect that the name has numbers and special characters in it and should notify the user and loop back to the customer name input		
3	Testing the input "Customer name"	3ewrew24f239047[[]'# Extreme	The program should detect that the name has numbers and special characters in it and should notify the user and loop back to the customer name input		
4	Testing the input "Customer's address"	22 Avenue Lane Normal	The program should save the address and continue through the program		
5	Testing the input "Customer's address"	3469 Avenue Lane Abnormal	The program should detect if the address has too many numbers or has any special characters that isn't supposed to be in an address, the program should notify the user and loop back to when the user needs to enter their address		
6	Testing the input "Customer's address"	127432 Av;;; lANE Extreme	The program should detect if the address has too many numbers or has any special characters that isn't supposed to be in an address, the program should notify the user and loop back to when the user needs to enter their address		
7	Testing the input "Phone number"	03844726559 Normal	The program should save the phone number and continue through the program		
8	Testing the input "Phone number"	02983374569332 Abnormal	The program should detect there are too many numbers in the phone number and should notify the user and loop back to the phone number input		
9	Testing the input "Phone number"	04826435hduydg'@] Extreme	The program should detect there are too many numbers and there are special characters in the		

			phone number and should notify the user and loop back to the phone number input		
10	Testing the input "Are these details correct?"	Y (Or) N Normal	The program should accept one of these options and either continue through the program or loop back to the start depending on the choice inputted		
11	Testing the input "Are these details correct?"	Yes (Or) (No) Abnormal	The program should notify the user that it is only accepting Y or N and loop back to "Are these details correct?"		
12	Testing the input "Are these details correct?"	Y3H (Or) N4H Extreme	The program should notify the user that it is only accepting Y or N and loop back to "Are these details correct?"		
13	Testing the input "Enter width of lawn"	23 Normal	The program should save the input and continue with the program		
14	Testing the input "Enter width of lawn"	98 Abnormal	The program should notify the user it is only accepting number between the range of 2-30 and loop back to the input		
15	Testing the input "Enter width of lawn"	204HJ Extreme	The program should notify the user it is only accepting numbers between the range of 2-30 and loop back to the input		
16	Testing the input "Enter length of lawn"	44 Normal	The program should save the input and continue the program		
17	Testing the input "Enter length of lawn"	67 Abnormal	The program should tell the user that it only accepts numbers between 2-50		
18	Testing the input "Enter length of lawn"	OSTUE7AC Extreme	The program should tell the user that it only accepts numbers between 2-50		

The plan shows testing of inputs however is not adequate as there is no testing of the final calculations or output of a correct bill. There is some testing of the logic. The plan is too narrow to confirm a working solution.

Band 1 (2 mark)

Activity 4 Code

```
'''This program perform basic mathematical calculation with data that user enters. Program is designed for gardener so that he can enter his
   details and customer's garden measurements. program also allow gardner to add special care if requested by customer. And after all program
   will generate a bill. program begins with user input where user have to enter customer name'''

while True:
   customer_name=input("Enter customer name- ")    #ask for name input
   if customer_name=="":
      print("Name can not be left blank!")       #error message if name is left blank
      continue
   if not customer_name.isalpha():              #check whether name contains any other value other then alphabetical value
      print("Name can only accept alphabetical value(can't accept any space value!)")
      continue
   if len(customer_name)>30:                    #check that name length is not exceeding 30 char
      print("Name length can't be more then 30 letters")
      continue
   else:
      break                                 #if name is valid break the loop


while True:
   customer_address=input("Enter customer address- ")     #ask for address input
   if customer_address=="":
      print("Address not be left blank!")                 #error message if address is left blank
      continue
   if len(customer_address)>50 or len(customer_address)<5:  #check that address is not exceeding given range
      print("Address length can't be more then 30 letters or less then 5 letters")
      continue
   else:
      break                                 #if address is valid break the loop

while True:
   customer_phone=input("Enter customer phone(add 0 in beginning)- ")
```

```
    if customer_phone=="":
        print("Phone number can not be left blank!")
        continue
    if not customer_phone.isdigit():
        print("Phone number can only accept numerical values")
        continue
    if len(customer_phone)!=11:
        print("Phone number must be 11 digit long")
        continue
    else:
        break


while True:
    lawn_length=input("Enter customer's lawn length- ")#ask user to enter lawn
length
    if lawn_length=="":
        print("Lawn length can not be left blank!")
        continue
    if not lawn_length.isdigit():
        print("Lawn length can only be a numerical values")
        continue
    if int(lawn_length)>50 or int(lawn_length)<2:          #check whether lawn
length is between given range or not
        print("Lawn length can only be between 2 and 50")
        continue
    else:
        lawn_length=int(lawn_length)
        break

while True:
    lawn_width=input("Enter customer's lawn width- ")      #ask user to enter
width length
    if lawn_width=="":
        print("Lawn width can not be left blank!")
        continue
    if not lawn_width.isdigit():
        print("Lawn width can only be a numerical values")
        continue
    if int(lawn_width)>30 or int(lawn_width)<2:    #check whether lawn width is
between given range or not
        print("Lawn width can only be between 2 and 30")
        continue
    else:
```

```
        lawn_width=int(lawn_width)
        break


lawn_area=lawn_length*lawn_width   #calculate the area of lawn based on
length and width entered by user
labour_cost=lawn_area*0.50         #standard labour cost is £0.50 per square
meter therefore total labour cost for whole area=lawn_area*0.50


print("\nWould you like to add any lawn care product(if NO press 4), We have
following lawn care product")
print("1.Luxury care\n2.Standard care\n3.Economy care\n4.Don't want to add
any additional product")#Prompt that will show user which care product he can
add
while True:
    selection=input("Enter number(int only): ")#user make a selection and
selection is stored in variable 'selection'
    if selection=="":
        print("Make a selection")
        continue
    if not selection.isdigit():
        print("Enter a numerical value please")
        continue
    if int(selection)<1 or int(selection)>4:
        print("Enter value between 1 and 4")
        continue
    if selection=='1':
        selection=1.15 #if user has selected option 1 that means he added luxury
service. So, we can simply say selection is 1.15(price of luxury care)
        break
    if selection=='2':
        selection=0.80#if user has selected option 2 that means he added
standard service. So, we can simply say selection is 0.80(price of standard care)
        break

    if selection=='3':
        selection=0.45#if user has selected option 3 that means he added
economy service. So, we can simply say selection is 0.45(price of eco care)
        break

    if selection=='4':#selection 4 means no product/service added so there will
be no cost
        selection=0
```

```
        break

    else:
        continue


print("\n\n----------Bill----------")
print("\nName",customer_name)
print("Addres",customer_address)
print("Phone",customer_phone)
print("\n\tLawn area-",lawn_area,"square meters")
print("\tLabour cost- £"+format(labour_cost,".2f"))
if selection==4:        #if selection was 4 that means no service/product was
added and we can simply show following messages
    print("\tCare cost-Not added")
    print("\tTotal cost exc vat- £"+labour)#total cost is equal to labour cost if no
service is been added
    print("\tTotal cost inc vat- £"+labour*1.2)#labour cost+20%VAT

else:       #contol only comes in else statement when selection wasn't 4 that
means special care was added so we also need to disply care cost.
    care_cost=lawn_area*selection
    total_exc_vat=care_cost+labour_cost
    total_inc_vat=total_exc_vat*1.2
    print("\tCare cost= £"+format(care_cost,".2f")) #care cost
    print("\tTotal cost exc vat- £"+format(total_exc_vat,".2f"))
    print("\tTotal cost inc vat- £"+format(total_inc_vat,".2f"))
```

The learner has produced a program that fully meets all the requirements. Accurate syntax and indentation have been used throughout the code and commenting is consistently clear and informative. Program outputs are accurate and informative, validation and other checks have been used which are all accurate, resulting in a largely robust program being created.

Band 4 (24 marks)

```
customer_name=input("What is the customer's name?\n")


customer_address=input("What is the customer's address?\n")#ask the user to
input the customer's address

while True:#starts a loop
    try:#attempts code within the loop
        customer_number=int(input("What is the customer's number?\n"))#ask the
user what the number is
        print("Acceptable data type")#prints acceptable number if it only consists of
numbers
        break;#ends the loop when an acceptable data type is reached
    except ValueError:#an exception within the loop that occurs when a ValueError
is reached
        print("only integers")#"only numbers" is printed when thier isn't an integer
stored in customer_number




while (customer_number/100000000000)>0.09 or
(customer_number/100000000000)<0.01: #starts a loop that checks if the
customers_number is bigger an 0.09 and smaller than 0.01 in which case it
wouldn't be a valid phone number
    print("number must contain 11 digits")
    customer_number=int(input("What is the customer's number?\n"))


width=float(input("What is the garden's width in meters?\n"))#asks the user to input
the garden's width as a float

while width<2 or width>30:
    print("Minimum 2 metres, maximum 30 metres")
    width=float(input("What is the garden's width in meters?\n"))#this loop continues
until the user inputs a width between with a minimum of 2 metres and a maximum
of 30 metres



length=float(input("What is the garden's length in meters?\n"))#asks the user to
input the garden's width as a float

while length<2 or length>50:
    print("Minimum 2 metres, maximum 50 metres")
    length=float(input("What is the garden's length in meters?\n"))#this loop
continues until the user inputs a length between with a minimum of 2 metres and a
maximum of 50 metres


garden_area=length*width#creates a variable called garden area which holds the
entire area of the garden
```

```
quality=input("What level of quality do you want, type l for luxury, s for standard, e for economy. If none of these are selected economy shall be the default choice.\n")#informs the user of the choicces to be made and what happens if they do not make the correct choice


product=0#holds value of the quality

if quality=="l":#if l is selected
        print("£1.15 per^2m ")
        product=1.15#variable that shalll hold the value of quality
elif quality=="s":
    print("£0.80 per^2m ")
    product=0.80#variable that shalll hold the value of quality
elif quality!="l" and quality!="s":
    print("£0.45 per^2m ")
    product=0.45#variable that shalll hold the value of quality


labour_cost=garden_area*0.5#variable that stores data of the labour cost

quality_cost=product*garden_area#variable that stores data of product cost per square metre

vat_cost=0.2*quality_cost*labour_cost#calculates vat cost

total_cost=vat_cost+quality_cost+labour_cost#calculates total cost


formatted_total="{:.2f}".format(total_cost)
formatted_width="{:.2f}".format(width)
formatted_length="{:.2f}".format(length)
formatted_area="{:.2f}".format(garden_area)
formatted_qualitycost="{:.2f}".format(quality_cost)#converts floats to 2 decimal places they are monetary value which is two dp
formatted_labour="{:.2f}".format(labour_cost)#converts floats to 2 decimal places they are monetary value which is two dp
formatted_vat="{:.2f}".format(vat_cost)#converts floats to 2 decimal places they are monetary value which is two dp


print("This is the toatal cost of the job is £"+ str(formatted_total))
print("The following reciept of the transaction.\nWidth="+(formatted_width)+"m")
print(("Length="+(formatted_length)+"m"))
print("Garden area=widthxlength")
print("Garden area="+(formatted_area)+"m")
print("Quality price:"+str(product))
print("Quality costs= product x garden area")
print("Quality cost:£"+(formatted_qualitycost))
print("Labour costs:£"+(formatted_labour))
```

```
print("vat costs= 20% of total costs")
print("Vat Cost:£"+(formatted_vat))
print("Total Cost"+ (formatted_total))#prints an entire recipt with the calculations formated to 2dp

print("Customer name:"+(customer_name))#outputs the name
print("Customer address:"+(customer_address))#ouytpusts the addess
print("Customer number: 0"+str(customer_number))#outputs number but with 0 at start so it appears like a regular number
```

The learner has produced a program that meets some of the requirements. There are some errors in the logic.
Code is efficient in some places and could be maintained, the program is functional. There are some mathematical errors in the calculations used for the outputs.

Band 2 (7 marks)

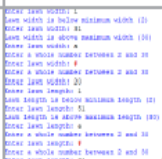
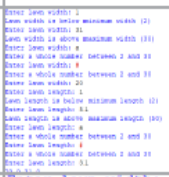
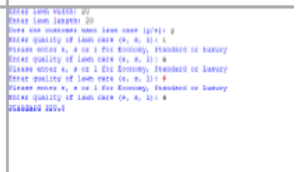
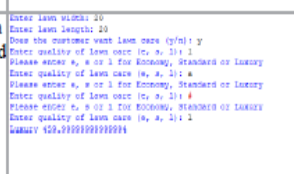
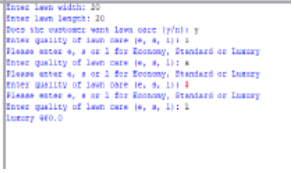
Activity 4 Testing

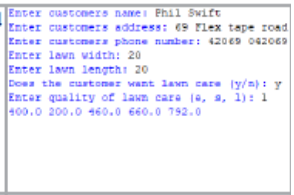
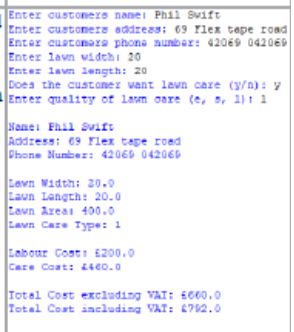
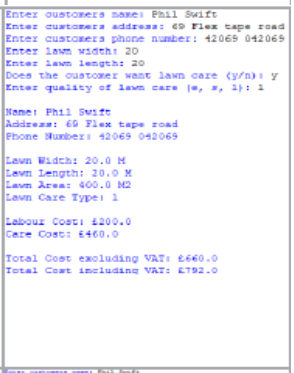
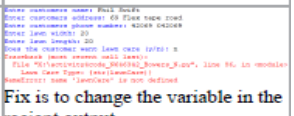
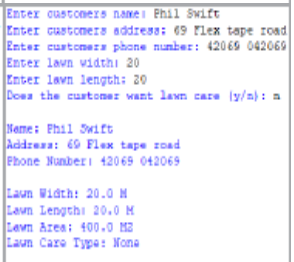
Document for Activities 3 and 4

Test Plan (add additional rows as required)

Program language the product is to be produced in (tick box for language used):

Python ☒ C Family ☐

Test Number	Purpose of test	Test Data	Expected Result	Actual Result	Comments
1	Check that the initial customer details have been stored correctly	Phil Swift, 69 Flex tape road, 42069 042069	The customers details should be printed out exactly how I entered them	Worked as intended	Enter customers name: Phil Swift Enter customers address: 69 Flex tape road Enter customers phone number: 42069 042069 Phil Swift 69 Flex tape road 42069 042069
2 and 3	Check that the lawn width and length error detection works as intended	1, 31, a, #, 20, 1, 51, a, #, 31	All should be rejected except for the 20 and the 31	Works as intended but there is a typo when the length is too big the maximum is stated to be 30 when it should be 50	 2 and 3 from the test plan have been combined into 2
2 and 3a	Check that the typo has been fixed	1, 31, a, #, 20, 1, 51, a, #, 31	All should be rejected except for the 20 and the 31 and the should now be a 50 instead of a 30 in the garden length size to big prompt	Fixed	
4	Check that the lawn area is calculated correctly	20, 20	The lawn area should be calculated to 400 square meters	Works as intended	Enter lawn width: 20 Enter lawn length: 20 400.0
5	Check that the lawn care selection works	1, #, a, n	The program should skip over the lawn selection process and put None in the care type and 0 as the care cost	Works as intended	Does the customer want lawn care (y/n): 1 Please enter y or n Does the customer want lawn care (y/n): # Please enter y or n Does the customer want lawn care (y/n): a Please enter y or n Does the customer want lawn care (y/n): n None 0.0
6	Check that the lawn care selection works	1, #, a, y	The program should then allow the user to select what type of lawn care to use	Works as intended	Does the customer want lawn care (y/n): 1 Please enter y or n Does the customer want lawn care (y/n): # Please enter y or n Does the customer want lawn care (y/n): a Please enter y or n Does the customer want lawn care (y/n): y Enter quality of lawn care (e, s, l):
7	Check that the lawn care type selection/ calculation works for economy care	1, a, #, e	The program should reject all but e and with a lawn area of 400 should give a price of 180 for lawn care	Works as intended	Enter lawn width: 20 Enter lawn length: 20 Does the customer want lawn care (y/n): y Enter quality of lawn care (e, s, l): 1 Please enter e, s or l for Economy, Standard or Luxury Enter quality of lawn care (e, s, l): a Please enter e, s or l for Economy, Standard or Luxury Enter quality of lawn care (e, s, l): # Please enter e, s or l for Economy, Standard or Luxury Enter quality of lawn care (e, s, l): e Economy 180.0
8	Check that the lawn care type selection/ calculation works for standard care	1, a, #, s	The program should reject all but s and with a lawn area of 400 should give a price of 320 for lawn care	Works as intended	
9	Check that the lawn care type selection/ calculation works for luxury care	1, a, #, l	The program should reject all but l and with a lawn area of 400 should give a price of 460 for lawn care	Works as intended but gives a recurring answer so will round to 2 decimal places	
9a	To check that the fix has worked	1, a, #, l	The program should now only give an answer of up to 2 decimal places	Works better than intended	

10	Check that the labour cost, total cost and total cost including vat are all calculated correctly	Phil Swift, 69 Flex tape road, 42069 042069, 20, 20, y, 1	The program should calculate the lawn area to be 400. The labour cost should be £200. The lawn care cost should be £460. The total cost excluding VAT should be £660 and the cost including VAT should be £792	Works as intended	
11	Check that the final output/receipt works as intended	Phil Swift, 69 Flex tape road, 42069 042069, 20, 20, y, 1	The final output should look like this: Name: Phil Swift Address: 69 Flex tape road Phone Number: 42069 042069 Lawn Width: 20 M Lawn Length: 20 M Lawn Area: 400 M2 Lawn Care Type: Luxury Labour Cost: £200 Lawn Care Cost: £460 Total cost excluding VAT: £660	Works as intended but forgot to add the units to the lawn width, length and area	
11a	Check that the added units to the receipt works	Phil Swift, 69 Flex tape road, 42069 042069, 20, 20, y, 1	The final output should look like this: Name: Phil Swift Address: 69 Flex tape road Phone Number: 42069 042069 Lawn Width: 20 M Lawn Length: 20 M Lawn Area: 400 M2 Lawn Care Type: Luxury Labour Cost: £200 Lawn Care Cost: £460 Total cost excluding VAT: £660 Total cost including VAT: £792	Works as intended	
11b	Check that not adding the option to have lawn care doesn't break the final receipt	Phil Swift, 69 Flex tape road, 42069 042069, 20, 20, n	The final output should look like this: Name: Phil Swift Address: 69 Flex tape road Phone Number: 42069 042069 Lawn Width: 20 M Lawn Length: 20 M Lawn Area: 400 M2 Lawn Care Type: None Labour Cost: £200 Lawn Care Cost: £0 Total cost excluding VAT: £200 Total cost including VAT: £240	Failed since I've been using the wrong variable for the Lawn care type on the receipt	 Fix is to change the variable in the reciept output
11c	Check that the fix has worked	Phil Swift, 69 Flex tape road, 42069 042069, 20, 20, n	The final output should look like this: Name: Phil Swift Address: 69 Flex tape road Phone Number: 42069 042069 Lawn Width: 20 M Lawn Length: 20 M Lawn Area: 400 M2 Lawn Care Type: None Labour Cost: £200 Lawn Care Cost: £0 Total cost excluding VAT: £200 Total cost including VAT: £240	Works as intended	

The learner has produced evidence of an iterative development process that identifies and resolves errors. Comments show understanding of the errors and how they were fixed. Comments and screen shots clearly show the process of testing and the output.

Band 3 (5 marks)

Document for Activities 3 and 4

Test Plan (add additional rows as required)

Program language the product is to be produced in (tick box for language used):

Python ■ C Family □

Test Number	Purpose of test	Test Data	Expected Result	Actual Result	Comments
1	Allow the user to enter a customer's name	Normal Jake	Take this variable down and output it at the end.	This works successfully	
2	Allow the user to enter a customer's name	Abnormal 3213131	This should print an error message and ask the question again.	Pauses my programme and says there is a syntax error	
2)A	Allow the user to enter a customer's name	Abnormal 3213131	This should print an error message and ask the question again.	This works successfully	The issue I had was I was missing a parentheses at the end of my line of code
3	Allows the user to enter a customer's address	Normal Ne11 abc	This should accept this variable and store it for when it needs to be outputted at the end	This works successfully	
4	Allows the user to enter a customer's address	Abnormal £££££££	This should print an error message and ask the user the question again	Nothing happens any my programme goes on and takes that input for an answer	
4)A	Allows the user to enter a customer's address	Abnormal £££££££	This should print an error message and ask the user the question again	Nothing happens any my programme goes on and takes that input for an answer	Still couldn't find a solution for it as it needs the correct usage of isalpha()
5	Allows the user to enter a customer's address	Extreme Ne1111111 hgjhkjh	As this is too long it should print an error message and ask the user the question again	Nothing happens any my programme goes on and takes that input for an answer	
5)A	Allows the user to enter a customer's address	Extreme Ne1111111 hgjhkjh	As this is too long it should print an error message and ask the user the question again	Nothing happens any my programme goes on and takes that input for an answer	This one I also couldn't find the correct usage of error handling. I need to set a maximum and minimum range but forgot where to put it.
6	Allow the user to enter a customer's phone number	Normal 01911 111111	This should accept the variable and store it so it can be outputted in the receipt at the end	This works successfully	
7	Allow the user to enter a customer's phone number	Abnormal £££££££	This should print an error message and ask the user the question again	This also works successfully my programme will only accept usage of numbers for this input	
8	Allow the user to enter a customer's phone number	Extreme 01111111 1111111	This should print an error message and ask the user the question again as the number is too long	My programme accepts this input and uses it throughout	
8)A	Allow the user to enter a customer's phone number	Extreme 01111111 1111111	This should print an error message and ask the user the question again as the number is too long	My programme accepts this input and uses it throughout	I need to again set a minimum and maximum range for this so I cannot do that
9	Ask the user what the length of their lawn is	Normal 5	This should accept the value and store it so it can be used in further calculations	This works successfully	
10	Ask the user what the length of their lawn is	Abnormal two	This should print an error message as it is	This just prints an error message and the	

			not in the format that the programme can read it	programme stops	
10)A	Ask the user what the length of their lawn is	Abnormal two	This should print an error message as it is not in the format that the programme can read it	This just prints an error message and the programme stops	Still couldn't find how to fix this I know what piece of code to use as I used it in my phone number piece of code however as I already have a different error handling function there I did not know how to add this one in
11	Ask the user what the length of their lawn is	Extreme 999999999	This should print an error message as there are far too many digits used and this is unrealistic	The programme does not accept this number	The reason for this however is the usage of my error handling function used for the range I have provided not to do with the input itself
12	The range for the lawn >50 or <2	Normal 5	This should accept the value and store it so it can be used in further calculations	This works successfully	
13	The range for the lawn >50 or <2	Abnormal five	As this is not in the correct format the programme should ask the question again	This does not work and just prints a red error message and the programme crashes	
13)A	The range for the lawn >50 or <2	Abnormal five	As this is not in the correct format the programme should ask the question again	This does not work and just prints a red error message and the programme crashes	Still could not find a solution for this I. I know there is a piece of code that allows you to convert a string to a float but I forgot what it was.
14	The range for the lawn >50 or <2	Extreme 51	Although this is just outside the barrier it should still ask the question again as the maximum range given is 50	This asks the user the question again until they put an answer within the given range	
15	Ask the user what the width of their lawn is	Normal 5	This should accept the value and store it so it can be used in further calculations	This works successfully	
16	Ask the user what the width of their lawn is	Abnormal two	This should print an error message as it is not in the format that the programme can read it	This does not work and just prints a red error message and the programme crashes	
16)A	Ask the user what the width of their lawn is	Abnormal two	This should print an error message as it is not in the format that the programme can read it	This does not work and just prints a red error message and the programme crashes	Still could not find a solution for this I. I know there is a piece of code that allows you to convert a string to a float but I forgot what it was.
17	Ask the user what the width of their lawn is	Extreme 999999999	This should print an error message as there are far too many digits used and this is unrealistic	The programme does not accept this number	The reason for this however is the usage of my error handling function used for the range I have provided not to do with the input itself
18	The range for the lawn >30 or <2	Normal 5	This should accept the value and store it so it can be used in further calculations	This works successfully	
19	The range for the lawn >30 or <2	Abnormal five	As this is not in the correct format the programme should ask the question	This does not work and just prints a red error message and the programme crashes	

			again		
19)A	The range for the lawn >30 or <2	Abnormal five	As this is not in the correct format the programme should ask the question again	This does not work and just prints a red error message and the programme crashes	Still could not find a solution for this I. I know there is a piece of code that allows you to convert a string to a float but I forgot what it was.
20	The range for the lawn >30 or <2	Extreme 31	Although this is just outside the barrier it should still ask the question again as the maximum range given is 30	This works successfully and asks the user the question over and over again until they give the correct answer	
21	The bill should be printed at the end	Prints automatically at the end	Everything should be on the bill correctly and separate bits of data should be on separate lines	All of this is working fine with no issues	

Testing shows evidence of a limited or linear development process, with minimal identification and resolution of errors.
To get into mark band 2 there must be evidence of errors and how they have been solved.

Band 1 (2 marks).

Activity 5

Evaluation

How well I met the success criteria of the scenario

When looking back at the success criteria we were given I feel like most if not all points that were given to me I have completed in some format. At the start we were asked to create a programme that would allow a user to enter a customer's name, address and phone number, then the length and width of the customer's lawn. After we were then asked to code it so it would allow the user to enter a lawn care quality if one was selected followed by the total cost of the job including lawn care and then a personalised bill everything on at the end. When you do one complete run through my programme you see that all these are there besides the showing of the calculations at the end on the bill. Now although this is very simple to complete and would not take much time it is a piece of information that I glanced over at the start of the exam and have forgot to add it in at the end. I have also added on the bill the cost without VAT and the cost with VAT included.

I feel like my programme is quite user friendly and simple and it provides accurate calculations. It also displays all numbers outputted in the pill to 2 decimal places so that it is in terms of money which also helps the user to understand.

The quality and performance of my program

When running through my program it is clear that the quality and performance are quite high. However there are many aspects of the piece of code that I have had to miss out due to lack of time I have had. There are many bits of error handling that I have had to miss out as they are usually quite lengthy bits of code and I just did not have time left at all. For example on the first input of my program where it asks the user to input his name. I would have liked to have added an extra few lines of code where it would have only allowed the user to input letters and anything else inputted would for the user to answer the question again until he inputted a piece of text that was valid. I would have done this by using the .isalpha() code however due to lack of time, I could not get this complete. I would have also like to do something similar for the range provided for the phone number and made it so no symbols could be added into the customers address variable. This similar to the customer's name issue is not too hard to solve however, just takes a lot of time. If I were to have had more time most issues to do with error handling would have been fixed as not only do they take a lot of time when adding to error handling functions into one input can sometimes be difficult and I would have to spend quite a bit of time using trial and error until I had it fixed.

The choices you made about coding conventions

With the use of coding conventions it allows your program to be much easier to read therefore making it user friendly. I have ensure that my variable names are specific so you know exactly what each variable is being used for and anyone would be able to pick the code up and understand what each one means. To make it even easier I have commented my program throughout so it's even more obvious what each section of the code is. This can take some time as you have to comment every separate section however, it pays of in the

long run as you could come back to the code and fully understand what each section is used for. I have also used snake case throughout my program as if I used multiple different ones (pascal and camel) it would start to get very confusing and would be far from user friendly. So sticking to one when declaring a variable makes it much more user friendly. In addition, indenting my code makes it look much neater and is much more simple to read over.

The changes you made during the development process

There were many issues that I ran into during the development process. Some I have managed to simply fix however, some took too long so I could not complete them. For example one issue I kept running into was I could not multiply a string by a float. This meant that I had to create another separate variable that I would declare as a float that would allow me to go ahead with my program. One other issue I ran into was not as obvious to find however when you try to crash the program it was appearing. On my inputs I did not have enough time to get all my error handling in however, at the start I had none. I saw that I had a little time left so I went back and added as much as I could before I ran out of time to move onto my next section. Although I did not have enough time to add all my error handling in I added as much as possible to make the program better.

The learner has demonstrated a mostly accurate and detailed understanding of technical concepts. Valid and mostly supported justification of coding conventions used, and the learner has made logical links between aspects of the solution and the requirements of the scenario. Valid and mostly supported judgements of the quality and performance of the program. Accurate technical vocabulary used to support arguments.

Band 3 (**9 marks**)

Evaluation

How well my solution meets the requirements of the scenario

I was tasked to create a program that would help calculate the cost of lawn cutting jobs.

I ensured I met the requirements of the client by focusing on the program being able to do the complex calculations to determine a final price as well as making sure I the program was simple enough for the user to understand.

In the program, at the start I made the program ask what the user's name, address and phone number was, I then made the program ask the client to confirm if the details inputted was correct. These are stored as variables for the bill to display

I then firstly asked the client what the width of the lawn was and made sure that the program would only accept numbers specified by the client. I then asked the client what the length of the lawn was and also made sure that the program would only accept numbers specified to the client. These are stored as variables for the bill to display

Next, I made the program give the user four options for the lawn care, as the client requested, these 4 options ranged from none to luxury, I also stored these as variables too.

I then make the program do some complex calculations in order to display the prices and stored variables onto the bill, making sure of little things such as the pricing being rounded to the next 2 decimal places.

The quality and performance of my program

During the development of the program, one of the big ideas was to focus on the overall performance of the program.

The program itself is efficient, the program uses a fairly small amount of code ad is procedurally driven, which means that the program will execute code the

form the very start to the very finish, this also makes it easier for a developer to read the code and understand it easily.

The program was most likely going to be used by customers, so I made the program as simple and readable as possible, with things such as precise constructions to help the user and a clear interface for the user to read, I also added a lot of error handling so that if the user makes a mistake, the user is notified and the program will not break.

The choices you made about coding conventions

One of the choices I made was the code to be readable and simple to follow, this would mean that anyone could be able to understand the code of the program, developer or not, I did this by adding text in the code which explains what that specific block or line of code does, I also spaced the code out in a way that the person who reads the code can see what parts do what. I also named things such as my variables to simple names such as customer_name or total_with_vat so that I and anyone else can understand what that variable contains.

The changes you made during the development process

I decided to not finish the pseudocode as I did not feel I had enough time to do, I felt that there were too many other things to do as well as the pseudocode.

Another change I decided to do during the development process was changing the length and width values to the correct numbers (as shown in the testing). This was so that I can fix the code for the client's requirements.

I did also change where the calculations part of the code was, before, it started at the very top of the code. I decided to change it because it would not be suitable for the simplicity and readability of the program.

The learner has demonstrated superficial understanding of relevant technical concepts. There is unsupported justification of changes made during the development process and limited justification of coding conventions supported.

Limited judgements about the quality and performance of the program keeps this evaluation in **mark band 1 (3 marks)**.

Summary

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

- Apply their knowledge to as many different scenarios as possible. The exam paper will always contain 5 activities which always be the same just the scenario would be different and therefore this will prepare learners to be able to provide answers to the given context under exam conditions.
- Use standard naming conventions throughout the design process and clearly demonstrate this in the flowchart and pseudocode.
- Pseudocode needs to be a detailed yet readable description of what a computer program must do, expressed in a natural language rather than in a programming language if top marks are to be achieved.
- Develop a better understanding of the testing process. Test plan must include normal, abnormal and extreme data. Testing must address errors encountered and how these were overcome. The testing must be iterative, document tests when code is being developed as this will give a true reflection of the development. If required by the scenario any final calculations and outputs from them should be tested and checked.
- Ensure the Program uses accurate validation and error checking procedures throughout, resulting in a robust program that minimises errors and handles unexpected events. This will enhance the completed solution and allow the higher mark bands to be accessed. Programs must address most requirements to gain higher marks.
- The evaluation needs to include a fully supported justification of changes made during the development process, as well as a fully supported justification of coding conventions selected if higher mark bands are to be accessed.