

Please check the examination details below before entering your candidate information

Candidate surname

Other names

Pearson BTEC Level 3
Nationals Extended
Certificate, Foundation
Diploma, Diploma,
Extended Diploma

Centre Number

--	--	--	--	--	--	--

Learner Registration Number

--	--	--	--	--	--	--	--	--	--

Friday 18 January 2019

Afternoon (Time: 2 hours)

Paper Reference **31768H**

Computing

Unit 1: Principles of Computer Science

You must have:

Information Booklet (enclosed)

Total Marks

--

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and learner registration number.
- Answer **all** questions.
- Answer the questions in the spaces provided
– *there may be more space than you need.*

Information

- The total mark for this paper is 90.
- The marks for **each** question are shown in brackets
– *use this as a guide as to how much time to spend on each question.*

Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers if you have time at the end.

Turn over ►

P51764A

©2019 Pearson Education Ltd.

1/1/1/1/1




Pearson

Answer ALL questions. Write your answers in the spaces provided.

Please refer to the Information Booklet in order to answer Question 1.

1 Jonty is a window cleaner. He would like a program to help manage his business.

Figure 1 in Section 1 of the Information Booklet shows some sample data about his customers.

(a) State **three** reasons why 'Telephone Number' would be a string data type instead of an integer.

(3)

Reason 1:

.....
.....

Reason 2:

.....
.....

Reason 3:

.....
.....

Jonty would like to offer customers over the age of 65 a 25% discount.

(b) Explain why Jonty would store a customer's date of birth rather than storing their age.

(2)

.....
.....
.....
.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



(c) Jonty would like to give each customer a status depending on the number of missing payments.

A customer's status is set to "Continue" when they have fewer than two missing payments. Otherwise the status is set to "Stop".

Complete this algorithm so that it meets these rules.

(2)

BEGIN

.....
.....

ELSE

status = "Stop"

ENDIF

END



Jonty will use functions in his programming code.

(d) Describe the difference between **declaring** a function and **calling** a function.

(4)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(e) Explain **one** benefit of using functions when creating program code.

(3)

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE




(f) Every morning Jonty will print a list of customers who will have their windows cleaned that day. They must live in the same area.

The requirements for his program are:

1. input an area number (e.g. 2)
2. display an error message if the area number is not between 1 and 7
3. search for customers who:
 - live in that particular area
 - have fewer than two missing payments
4. print the details of each customer who meets both of these criteria
5. it should repeat steps 3 and 4 until all customers have been checked.

Draw a **flowchart** that meets the requirements for the algorithm.

(8)



Continue your flowchart on the next page.

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

[Large empty rectangular area for writing answers]

(Total for Question 1 = 22 marks)



2 Curtis would like to lead a healthier lifestyle. He would like a program to monitor how much time he spends exercising each day.

(a) The program code will contain different procedures.

Explain what is meant by the term 'procedure' when creating program code.

(3)

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



(b) Curtis will use a password to access his program. He has written an algorithm to compare a stored password with a user input. This is written in pseudocode.

```
BEGIN  
storedPassword = ComputerScience1#  
INPUT enteredPassword  
FOR count = 1 TO 3  
    IF storedPassword = enteredPassword THEN  
        BREAK  
    ELSE  
        INPUT enteredPassword  
    ENDIF  
NEXT count  
ENDFOR  
END
```

Explain the relationship between the **FOR** loop and **BREAK** statement in the pseudocode.

(4)

.....

.....

.....

.....

.....

.....

.....

.....



(c) Curtis would like to expand his program to make sure a new password is a minimum length.

The rules for the algorithm are:

1. check the password is at least eight characters long
2. only continue if the password is of the correct length
3. when the password is the correct length print "Accepted".

Develop an algorithm that meets the given rules.

Write your answer using pseudocode.

(4)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



Curtis has recorded the number of minutes he has exercised each day over a seven-day period.

He wants to sort this data into order from lowest to highest. He will use a standard sorting algorithm.

- (d) Demonstrate how a **quick sort** can put the data in order from the lowest number to the highest number.

Use the **left-most value** for the pivots.

(5)

44	75	23	12	55	43	33
----	----	----	----	----	----	----

You should continue your quick sort on the next page.

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

Large empty rectangular area for writing or drawing.



Curtis could use a bubble sort to put the data in order from the lowest number to the highest number.

The exercise times are shown again here for your reference.

44	75	23	12	55	43	33
----	----	----	----	----	----	----

(e) Describe how a bubble sort would use **loops** to sort the exercise times into order. (4)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(Total for Question 2 = 20 marks)

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



Please refer to Section 2 of the Information Booklet in order to answer Questions 3(b) and 3(c).

3 Theo owns a restaurant. He would like a program to manage his table bookings. The programming code will be written using an object-oriented programming language.

(a) Explain what is meant by the term 'encapsulation' when creating program code.

(4)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

Four sets of horizontal dotted lines for writing, each set consisting of a solid top line, a dotted middle line, and a solid bottom line.



(c) **Figure 2b** shows the plan of Theo's restaurant and the rules that are followed to find a table.

Theo will structure his code using a combination of sequence, conditional and iterative control structures.

Analyse how Theo could use these control structures in his program code to meet the rules given in **Figure 2b**.

(10)

Area with horizontal dotted lines for writing the answer.

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

Handwriting practice area with 10 horizontal dotted lines.

(Total for Question 3 = 22 marks)



(c) Evaluate the extent to which the algorithm meets the algorithm rules.

Your evaluation should cover the effectiveness and accuracy of the algorithm.

(12)

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

Area with horizontal dotted lines for writing.

(Total for Question 4 = 26 marks)

TOTAL FOR PAPER = 90 MARKS



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



Pearson BTEC Level 3 Nationals Extended Certificate
Foundation Diploma, Diploma, Extended Diploma

Friday 18 January 2019

Afternoon

Paper Reference **31768H**

Computing

Unit 1: Principles of Computer Science

Information Booklet

Instructions

- You will need the information in this booklet to answer some questions.
- Read the information carefully.
- You must **not** write your answers in this booklet.
- Only your answers given on the question paper booklet will be marked.
- Do not return this Information Booklet with the question paper.

Turn over ►

P51764A

©2019 Pearson Education Ltd.

1/1/1/1




Pearson

SECTION 1

The information in this section should be used to answer Question 1.

Figure 1 shows some sample data about Jonty's customers.

ID	Area Number	House Number	Road Name	First Name	Surname	Date of Birth	Telephone Number	Missing Payments
1	1	128	Liverpool Road	Betty	Hepworth	19/01/1986	(01704) 211 988	0
2	2	89	Ashley Street	Chloe	Birks	01/06/1975	(01704) 221 155	2
3	5	72	Duke Street	Kian	Miller	25/03/1961	(01704) 236 664	1
4	6	14	Brade Street	Charlie	Brown	09/10/1989	(0151) 278 884	1
5	1	16	Lexton Drive	Emma	Smith	11/04/1990	(01704) 222 123	2
6	1	17a	Preston Road	Chris	Johnson	19/12/1979	(01704) 266 655	0
7	7	137	Lord Street	Lucy	Malone	19/08/1981	(0151) 211 113	1

Figure 1

Jonty allocates an Area Number to each customer. Houses in the same area are cleaned on the same day.

This shows how many payments are missing for each customer.

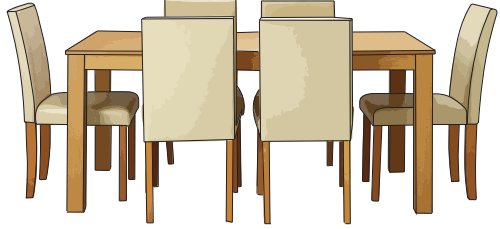
BLANK PAGE

Section 2

The information in this section should be used to answer Question 3.

Figure 2a shows the design of the screen that will be used to enter the details of the tables booked by customers in Theo's restaurant.

Find a table



Ticks and crosses are used to confirm successful data entry. **Other methods** can also be used to confirm this.

Surname ✓

Arrival Time ✓

Arrival Date ✗

Seating Preference ✓

Find Table

The first letter should be capitalised. The following letters should be in lower case.

The time should be expressed as **HH : MM**

The date should be expressed as **DD / MM / YY**

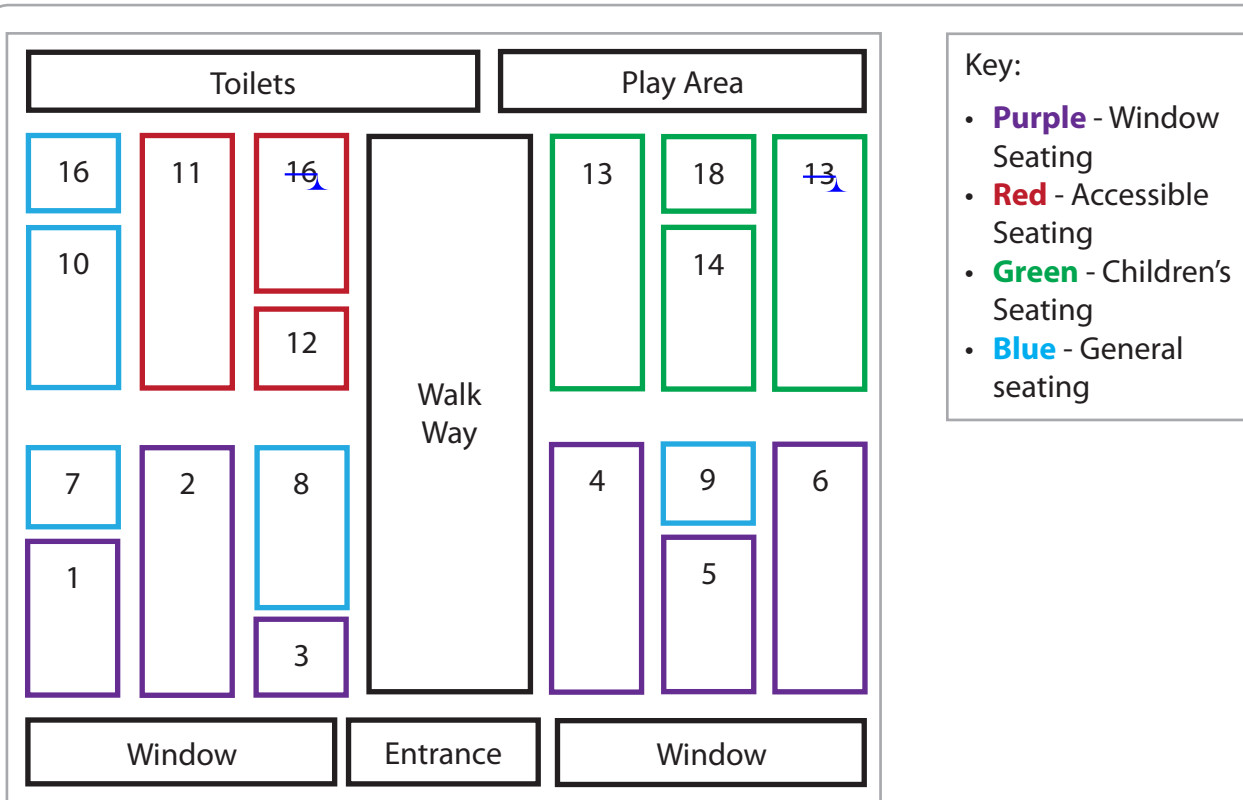
This can be in the:

- Window Seating
- Accessible Seating
- Children's Seating
- General Seating – This will be automatically selected if no option is selected.

When pressed this button will run validation rules and search for a table.

Figure 2a

Figure 2b shows the plan of the restaurant and rules that are followed to find a table.



1. A check will be made to make sure all booking information is entered:

- Surname
- Arrival Time
- Arrival Date
- Seating Preference

2. When the booking information is entered these rules are used to find a suitable table:

- **Rule 1** – Table in the seating area requested must be available at the time of arrival for 90 minutes.
- **Rule 2** – When rule 1 is not met, the customer is asked to change their arrival time. A check is made to see if a table in the seating area requested is available at the new arrival time for 90 minutes.
- **Rule 3** – When rule 1 and rule 2 are not met a check is made to see if a table in any seating area is available for 90 minutes.

If no tables are found the customer is told that no tables are available.

Figure 2b

Section 3

The information in this section should be used to answer Question 4.

Figure 3a shows the first design for the main menu for Poppy's revision program.

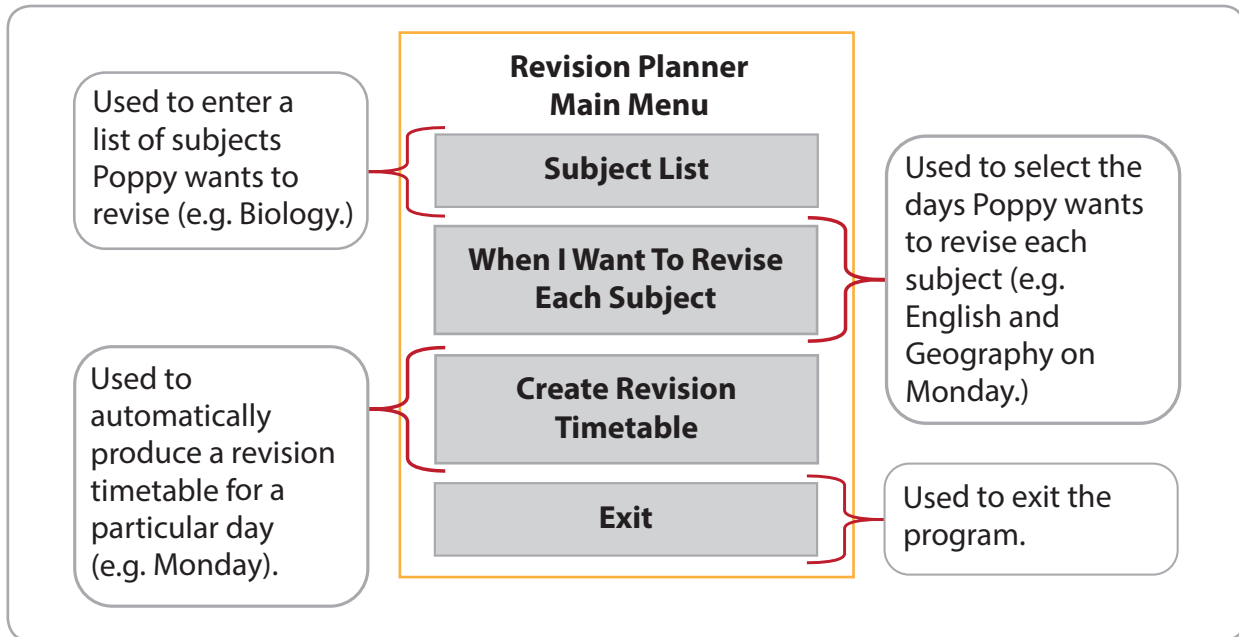


Figure 3a

BLANK PAGE

Figure 3b shows the rules that are followed when the 'Create Revision Timetable' button is pressed.

Algorithm Rules

1. The day is input (e.g. Monday).
2. The start time is input (e.g. 17:30).
3. The number of subjects to be revised that day is counted.
4. Each subject must be revised for 60 minutes. The revision time is calculated by multiplying the number of subjects by 60.
5. There should be a rest break every 45 minutes. The number of rest breaks is calculated by dividing the revision time by 45.
6. A rest break lasts for 20 minutes. The time required for rest breaks is calculated by multiplying the number of rest breaks by 20.
7. The total time is then calculated by adding the revision time to the rest break time.
8. The program should then print the:
 - number of subjects to be revised that day
 - revision start time
 - revision end time
 - number of breaks needed.

Figure 3b

Figure 3c shows an algorithm that will produce a revision timetable.

It is written in pseudocode.

```
01 BEGIN
02 //SECTION A – DAYS AND THE SUBJECTS THAT WILL BE REVISED
03 monday = [Biology, Computing, History]
04 tuesday = [Chemistry, Maths]
05 wednesday = [Geography, French, Physics]
06 thursday = [English, Maths]
07 friday = [Computing, History]
08 saturday = [Biology, Chemistry, Physics, Maths]
09 sunday = [Geography, French]
10
11 //SECTION B - VARIABLE DECLARATIONS
12 revisionDay = x
13 startTime = x
14 numberOfSubjects = 0
15 revisionTime = 0
16 numberOfBreaks = 0
17 breakTime = 0
18 totalTime = 0
19
20 //SECTION C - CALCULATING THE NUMBER OF SUBJECTS THAT NEEDS REVISING
21 INPUT revisionDay
22 INPUT startTime
23 FOR all items in revisionDay
24     numberOfSubjects = numberOfSubjects +1
25 ENDFOR
26
27 //SECTION D - CALCULATING THE REVISION TIME
28 revisionTime = numberOfSubjects/60
29
30 //SECTION E - CALCULATING THE NUMBER OF REST BREAKS AND REST TIME
31 WHILE revisionTime > 45 THEN
32     numberOfBreaks = numberOfBreaks +1
33 ENDWHILE
34 breakTime = numberOfBreaks *20
35
36 //SECTION F - CALCULATING THE TOTAL TIME
37 totalTime = revisionTime + breakTime
38
39 //SECTION G – PRINTING INFORMATION TO THE USER
40 PRINT "The number of subjects that need revising is" + numberOfSubjects
41 PRINT "You will start revision at" + startTime
42 PRINT "You will stop revising at" + (startTime+totalTime)
43
44 END
```

Figure 3c



BLANK PAGE



BLANK PAGE



BLANK PAGE

