Write your name here

Surname	Other names

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

Centre Number

Learner Registration Number

Computing

Unit 1: Principles of Computer Science

Wednesday 17 May 2017 – Morning
Time: 2 hours

Paper Reference
31768H

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 ▶

P51741A

©2017 Pearson Education Ltd.

1/1/1/1/1/1/1/1

P 5 1 7 4 1 A 0 1 2 0

Pearson

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

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

1 Rashida currently uses a paper-based system to manage a small independent gym. She wants a program to replace the paper-based system.

Section 1 of the Information Booklet contains information about the gym and the screen design for the new program.

The algorithm that calculates the member discount based on the member age is shown here.

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

(3)

```
BEGIN

INPUT age

[IF]  age >=14 AND <=16:
      Discount = 30%

[   ] age =17 [   ] =18:
      Discount = 20%

[ELIF] age >= 50:
       Discount = 40%

[   ] : Discount = 0%

END
```

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

(b) The Python code shown has a variable called 'age'.

```
age = input ("Please Enter Your Age")
if age >= 14 and age <=16:
    Discount = 30
```

When 'age' is input by the user it is stored as a string data type.

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

(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

Figure 1b shows the screen design for the new program. The programmer will create the screen using an event driven programming language.

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

(6)

Event 1

Event handler 1

Event 2

Event handler 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

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

(4)

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

(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

(f) Member details are stored in a record.

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

(3)

(Total for Question 1 = 22 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

2 Sally is currently investigating possible bank accounts for her savings. Sally would like a program that will help her choose which account will pay the most interest on her savings.

(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.	
Only accept the first two digits entered for the interest rate.	
Only allow a number between 1 and 20 to be entered for the interest rate.	

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

Sally's requirements for the program are given in the table.

1	The number of bank accounts to be compared is entered and stored as a variable.
2	The money to be saved is entered and stored as a variable.
3	The interest rate is then entered and stored as a variable.
4	The interest is calculated by dividing the money to be saved by 100 and then multiplying this by the interest rate.
5	The total is calculated by adding the money to be saved to the interest.
6	The total is then output.
7	The program should loop for the entered number of bank accounts.

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) Create an algorithm, using a **flow chart**, that meets Sally's requirements.

(6)

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) 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)

(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)

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

(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)

(Total for Question 2 = 22 marks)

Please refer to the Inforation Booklet in order to answer Question 3.

3 When new students arrive at school they need to be given a username to access the school computer network. A technician would like to import students' details into a program from a text file to create a username automatically.

The text file and the rules that are used to generate usernames can be seen in Section 2 of the Information Booklet.

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

(8)

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) 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)

(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)

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

(Total for Question 3 = 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 the Information Booklet in order to answer Question 4.

4 Richard owns a lift maintenance business. When work has been carried out, Richard tests the lift to ensure it is working correctly. He uses a program to keep these test logs organised.

Each lift has a target time that indicates the maximum amount of time a lift should take to travel from the ground floor to the top floor. If the actual time taken does not match the target time it could indicate that there is a problem.

(a) **Figure 3a** shows the screen layout for the main menu. **Figure 3b** shows part of the programming code.

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

(6)

Error 1

Line number

.....

Description

.....

.....

.....

.....

.....

Error 2

Line number

.....

Description

.....

.....

.....

.....

.....

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) **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)

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

(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)

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

(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