

Pearson BTEC Level 3 Nationals

Computing

Unit 1: Principles of Computer Science

Wednesday 17 May 2017 – Morning

Information Booklet

Paper Reference

31768H

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.

Turn over ►

P51741A

©2017 Pearson Education Ltd.

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



Pearson

SECTION 1

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

Figure 1a shows some information about the current system used by Rashida.

Membership

- When new members join, their personal details are entered into a paper-based folder and staff have to manually search this folder if member details need updating.
- Their membership expiry date is calculated manually and entered into the diary. Staff have to manually check when memberships are due to expire.
- When memberships expire, members must then return their membership cards as they could continue to access the gym's facilities without paying.

Types

- There are two types of membership, which allocate time restrictions when members can access the facilities.
- Occasionally members still gain access to the gym beyond their time restriction.

Discounts

- Some members receive a discount.
 1. Members who are **aged between 14 and 16** receive a 30% discount.
 2. Members who are **aged between 17 and 18** receive a 20% discount.
 3. Members who are **aged 50 and over** receive a 40% discount.
 4. All other members receive no discount.
- When the discount amount has been set, this is not updated automatically. This means that discounts are often applied incorrectly.

Figure 1a

Figure 1b shows a screen design for the new program for Rashida and example data.

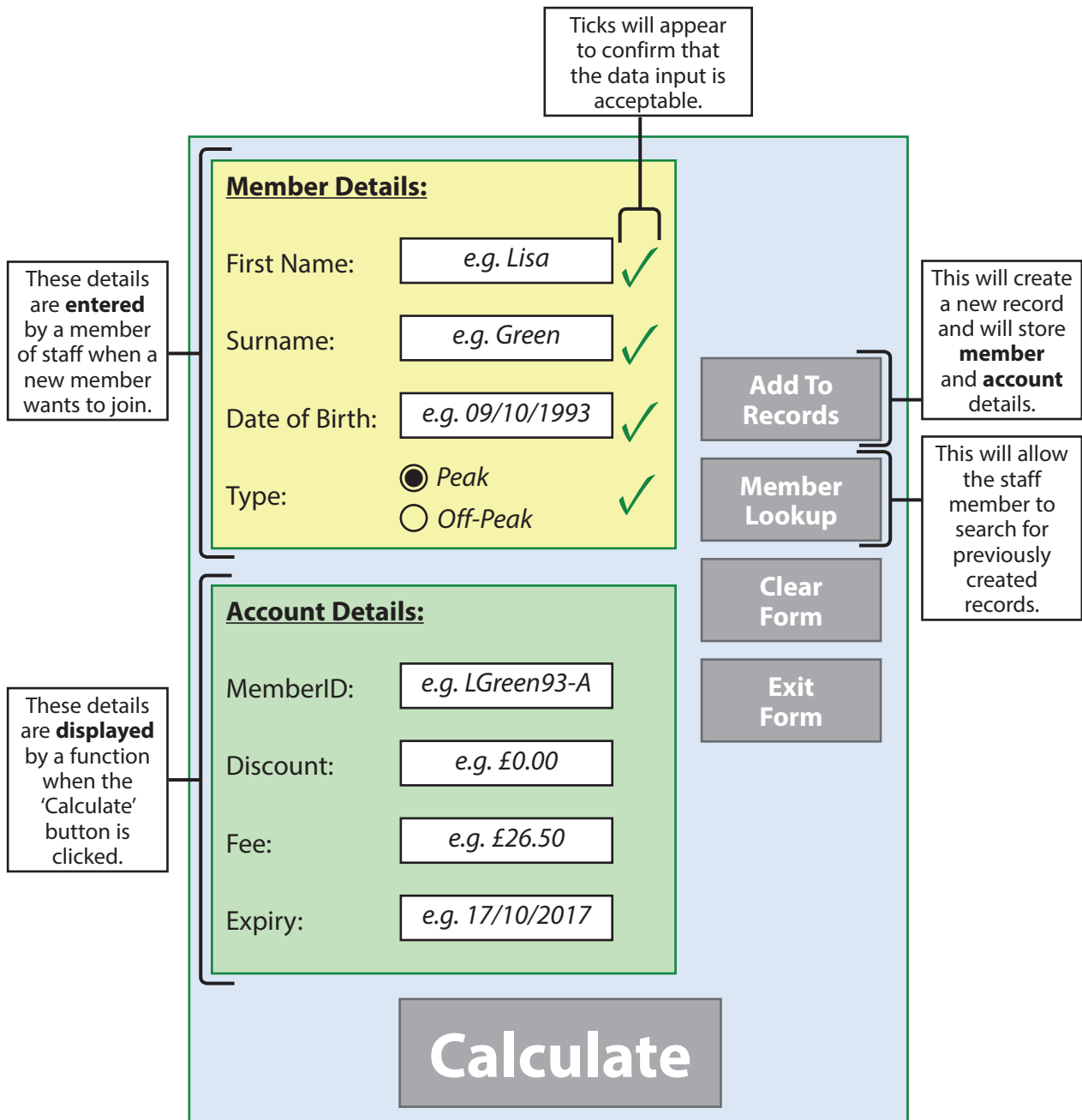


Figure 1b

SECTION 2

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

Figure 2 shows the file that is used by the technician to create new usernames.



Figure 2

Rules for generating a new username

- A username must be made up from:
 1. The year they started (e.g. 2017)
 2. The first letter of their first name (e.g. A)
 3. Their surname (e.g. Cain)
 4. A number 1.
- If the username that has been created already exists then it must increase the number at the end of the username by one to make it unique. (e.g. 2017ACain2)
- The username should then be printed to the screen.

SECTION 3

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

Figure 3a shows the main menu for the program used by Richard.



Figure 3a

Figure 3b shows the programming code for the main menu. It is written using C#.

```
55 }
56 void New_logClick(object sender, EventArgs e)
57 {
58     New_Log newLogForm = new New_Log();
59     newLogForm.Hide();
60 }
61 void View_logClick(object sender, EventArgs e)
62 {
63
64     View_Log viewLogForm = new View_Log();
65     viewLogForm.Show();
66 }
67 void View_faultsClick(object sender, EventArgs e)
68 {
69     View_Fault viewFaultForm = new View_Fault();
70     viewFaultForm.Show();
71 }
72 void QuitClick(object sender, EventArgs e)
73 {
74     Application.Run();
```

Figure 3b

Figure 3c shows the screen used by Richard to create a new test log.

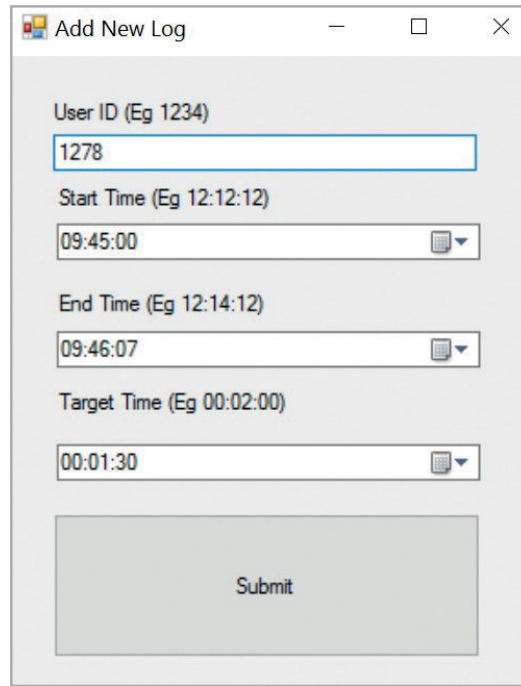


Figure 3c

Figure 3d shows the programming code used on the test log screen. It is written using C#.

```
1  using System;
2  using System.Drawing;
3  using System.Windows.Forms;
4  using System.IO;
5
6  namespace Lifttest_c_sharp
7  {
8      public partial class New_Log : Form
9      {
10         public New_Log()
11         {
12             InitializeComponent();
13         }
14         void SubmitClick(object sender, EventArgs e)
15         {
16             if (userid.Text != "" && ridename.Text != ""){
17
18                 string lineCount = File.ReadAllLines(@".\log.txt").Length.ToString();
19
20                 string logID = lineCount + userid.Text + starttime.Text + ","
21                     + endtime.Text + "," + targettime.Text + Environment.NewLine;
22
23                 File.AppendAllText(".\log.txt", log);
24
25                 TimeSpan start_time = TimeSpan.Parse(starttime.Text);
26                 TimeSpan end_time = TimeSpan.Parse(endtime.Text);
27                 TimeSpan target_time = TimeSpan.Parse(targettime.Text);
28                 TimeSpan time_taken = end_time - start_time;
29
30                 if(time_taken > target_time){
31
32                     MessageBox.Show("Time taken exceeds target time. A fault will be logged");
33
34                     string faultlog = logID + "," + userid.Text + "," + "," + starttime.Text + ","
35                         + endtime.Text + "," + targettime.Text + "," + "Maintenance Check" + Environment.NewLine;
36
37                     File.AppendAllText(".\fault.txt", faultlog);
38                 }
39                 this.Close();
40             } else {
41                 MessageBox.Show("Enter a valid User ID");
42             }
43         }
44     }
45 }
46
```

Figure 3d

BLANK PAGE

BLANK PAGE