

OXFORD

INTERNATIONAL
AQA EXAMINATIONS

INTERNATIONAL GCSE

COMPUTER SCIENCE

9210

Paper 1 Programming

Mark scheme

June 2021

Version: 1.0 Final



2 1 6 Y 9 2 1 0 1 / M S

Mark schemes are prepared by the Lead Assessment Writer and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation events which all associates participate in and is the scheme which was used by them in this examination. The standardisation process ensures that the mark scheme covers the students' responses to questions and that every associate understands and applies it in the same correct way. As preparation for standardisation each associate analyses a number of students' scripts. Alternative answers not already covered by the mark scheme are discussed and legislated for. If, after the standardisation process, associates encounter unusual answers which have not been raised they are required to refer these to the Lead Examiner.

It must be stressed that a mark scheme is a working document, in many cases further developed and expanded on the basis of students' reactions to a particular paper. Assumptions about future mark schemes on the basis of one year's document should be avoided; whilst the guiding principles of assessment remain constant, details will change, depending on the content of a particular examination paper.

Further copies of this mark scheme are available from [oxfordaqaexams.org.uk](https://www.oxfordaqaexams.org.uk)

Copyright information

OxfordAQA retains the copyright on all its publications. However, registered schools/colleges for OxfordAQA are permitted to copy material from this booklet for their own internal use, with the following important exception: OxfordAQA cannot give permission to schools/colleges to photocopy any material that is acknowledged to a third party even for internal use within the centre.

Copyright © 2023 Oxford International AQA Examinations and its licensors. All rights reserved.

Section A

Question	Part	Marking guidance	Total marks
01	1	EnterWeight / CalculateCalories / EnterDistance;	1 AO3=1

Question	Part	Marking guidance	Total marks
01	2	PersonData; Exercise;	2 AO3=2

Question	Part	Marking guidance	Total marks
01	3	(1D) array/list; of type Exercise;	2 AO3=2

Question	Part	Marking guidance	Total marks
02	1	3;	1 AO3=1

Question	Part	Marking guidance	Total marks
02	2	FOR//FOR EACH; A. for Activity in ActivityLog: A. For Each Activity In ActivityLog A foreach (Exercise Activity in ActivityLog) I. case and spacing	1 AO3=1

Question	Part	Marking guidance	Total marks
02	3	Checks if the activities are the same; It converts the type of activity in both strings to lower case // ensuring the comparison is independent of case;	2 AO3=2

Question	Part	Marking guidance	Total marks
03	1	100;	1 AO3=1

Question	Part	Marking guidance	Total marks
----------	------	------------------	-------------

03	2	1 mark: The outer/while loop would stop repeating if no swaps were made 1 expansion mark: so not as many iterations would be needed (if the list is partially sorted) // this is because if a pass is made with no swaps the list must be sorted 1 mark: The inner/for loop iterates one less time 1 expansion mark: each time the outer/while loop iterates // this is because each time the outer/while loop iterates one additional item will be in the correct place If no other marks awarded, award 1 mark for recognising there will be fewer iterations or comparisons	4 AO3=4
----	---	--	------------

Question	Part	Marking guidance	Total marks
----------	------	------------------	-------------

03	3	Temp is used to ensure that the items/activities at Index and Index + 1 are swapped; without items/activity at Index / Index + 1 being duplicated/overwritten; // If this was not used then there would end up being two copies of one activity/item; and one activity would be removed from /not be in the ActivityLog;	2 AO3=2
----	---	---	------------

Question	Part	Marking guidance	Total marks
----------	------	------------------	-------------

03	4	10;	1 AO3=1
----	---	-----	------------

Question	Part	Marking guidance	Total marks
----------	------	------------------	-------------

03	5	When the array / list / ActivityLog is already sorted; in ascending order // by distance; R. descending order	2 AO3=2
----	---	--	------------

Question	Part	Marking guidance	Total marks
----------	------	------------------	-------------

04	1	9; 101;	2 AO3=2
----	---	------------	----------------

Question	Part	Marking guidance	Total marks
----------	------	------------------	-------------

04	2	67 / 95;	1 AO3=1
----	---	----------	----------------

Section B

Question	Part	Marking guidance	Total marks
05	1	1 mark: Program displays correct message. A. minor typos and incorrect case. I. full stop at end of message 1 mark: Statements are at a position so that message is displayed whenever the menu is displayed and before the current menu title MAIN MENU; 1 mark: Blank line is printed after the message is displayed; Python <pre>def DisplayMenu(): print("Personal Fitness Application") print() print("MAIN MENU") print() print("1. Display User Information") print("2. Load Activity Log File") print("3. Add Activity") print("4. Display Activities") print("5. Sort Activities by distance") print("Q. Quit") print()</pre> C# <pre>private static void DisplayMenu() { Console.WriteLine("Personal Fitness Application"); Console.WriteLine(); Console.WriteLine("MAIN MENU"); Console.WriteLine(); Console.WriteLine("1. Display User Information"); Console.WriteLine("2. Load Activity Log File"); Console.WriteLine("3. Add Activity"); Console.WriteLine("4. Display Activity Details"); Console.WriteLine("5. Sort Activities by Distance"); Console.WriteLine("Q. Quit"); Console.WriteLine(); }</pre> VB <pre>Sub DisplayMenu() Console.WriteLine("Personal Fitness Application") Console.WriteLine() Console.WriteLine("MAIN MENU") Console.WriteLine() Console.WriteLine("1. Display User Information") Console.WriteLine("2. Load Activity Log File") Console.WriteLine("3. Add Activity") Console.WriteLine("4. Display Activity Details") Console.WriteLine("5. Sort Activities by Distance") Console.WriteLine("Q. Quit") Console.WriteLine() End Sub</pre>	3 AO3=3

Question	Part	Marking guidance	Total marks
05	2	Screen capture(s) must match code from Q5.1 1 mark: Correct message displayed above menu. A. Incorrect message if matches Q5.1 Personal Fitness Application MAIN MENU 1. Display User Information 2. Load Activity Log File 3. Add Activity 4. Display Activities 5. Sort Activities by Distance Q. Quit Please enter your choice:	1 AO3=1

Question	Part	Marking guidance	Total marks
06	1	1 mark: Else or other mechanism used to display message under correct conditions 1 mark: Program displays correct message A. minor typos and incorrect case. I. full stop at end of message 1 mark: Statements are at a position so that message is displayed whenever an invalid menu option is entered. 1 mark: The way the other parts of the selection structure operate are not affected. Python <pre>def Main(): Individual = PersonData("1001", "NameOne", "M", 21, 70.2) Choice = input("Enter D to use the default person data: ") if Choice != 'D': Individual = EnterUserData(Individual) ActivityLog=[] MenuOption = "0" while MenuOption != "Q": DisplayMenu() MenuOption = GetChoice() if MenuOption == "1": DisplayUserData(Individual) elif MenuOption == "2": ActivityLog = LoadActivity(Individual, ActivityLog) elif MenuOption == "3": ActivityLog = AddActivity(Individual, ActivityLog) elif MenuOption == "4": TypeOfActivity = input("What type of activity do you want to display: ") DisplayActivity(ActivityLog, TypeOfActivity) elif MenuOption == "5": ActivityLog = SortByDistance(ActivityLog) elif MenuOption == "Q": print("Goodbye, thank you for using this application.") else: print("Invalid Menu Option")</pre> C# <pre>static void Main(string[] args) { PersonData Individual; Individual.ID = "1001"; Individual.Name = "NameOne"; Individual.Gender = "M"; Individual.Age = 21; Individual.Weight = 70.2; Console.WriteLine("Enter D to use the default person data:</pre>	4 AO3=4

```
");
string Choice = Console.ReadLine();
if (Choice != "D")
{
    Individual = EnterUserData(Individual);
}
List<Exercise> ActivityLog = new List<Exercise>();
string MenuOption = "0";
while (MenuOption != "Q")
{
    DisplayMenu();
    MenuOption = GetChoice();
    if (MenuOption == "1")
    {
        DisplayUserData(Individual);
    }
    else if (MenuOption == "2")
    {
        ActivityLog = LoadActivity(Individual,
ActivityLog);
    }
    else if (MenuOption == "3")
    {
        ActivityLog = AddActivity(Individual, ActivityLog);
    }
    else if (MenuOption == "4")
    {
        Console.WriteLine("What type of activity do you want
to display: ");
        string TypeOfActivity = Console.ReadLine();
        DisplayActivity(ActivityLog, TypeOfActivity,
Individual);
    }
    else if (MenuOption == "5")
    {
        ActivityLog = SortByDistance(ActivityLog);
    }
    else if (MenuOption == "Q")
    {
        Console.WriteLine("Goodbye, thank you for using
this application.");
    }
    else
    {
        Console.WriteLine("Invalid Menu Option");
    }
}
}
```

		VB <pre> Sub Main() Dim Individual As PersonData = New PersonData() Individual.ID = "1001" Individual.Name = "NameOne" Individual.Gender = "M" Individual.Age = 21 Individual.Weight = 70.2 Console.WriteLine("Enter D to use the default person data: ") Dim Choice As String = Console.ReadLine() If Choice <> "D" Then Individual = EnterUserData(Individual) End If Dim ActivityLog As New List(Of Exercise) Dim MenuOption As String = "0" While MenuOption <> "Q" DisplayMenu() MenuOption = GetChoice() If MenuOption = "1" Then DisplayUserData(Individual) ElseIf MenuOption = "2" Then ActivityLog = LoadActivity(Individual, ActivityLog) ElseIf MenuOption = "3" Then ActivityLog = AddActivity(Individual, ActivityLog) ElseIf MenuOption = "4" Then Console.WriteLine("What type of activity do you want to display: ") Dim TypeOfActivity As String = Console.ReadLine() DisplayActivity(ActivityLog, TypeOfActivity, Individual) ElseIf MenuOption = "5" Then ActivityLog = SortByDistance(ActivityLog) ElseIf MenuOption = "Q" Then Console.WriteLine("Goodbye, thank you for using this application.") Else Console.WriteLine("Invalid Menu Option") End If End While End Sub </pre>	
--	--	--	--

Question	Part	Marking guidance	Total marks
06	2	Screen capture(s) must match code from Q6.1 1 mark: Correct error message is displayed when invalid menu option is entered. A. Incorrect error message if matches Q6.1 Personal Fitness Application MAIN MENU 1. Display User Information 2. Load Activity Log File 3. Add Activity 4. Display Activities 5. Sort Activities by Distance Q. Quit Please enter your choice: A Invalid Menu Option Personal Fitness Application MAIN MENU 1. Display User Information 2. Load Activity Log File 3. Add Activity 4. Display Activities 5. Sort Activities by Distance Q. Quit Please enter your choice: -1 Invalid Menu Option Personal Fitness Application	1 AO3=1

Question	Part	Marking guidance	Total marks
07	1	1 mark: DisplayMenu subroutine includes the new menu option 6. Change Individual Person Data 1 mark: case of output message. A. minor typos in output message. 1 mark: spacing in output message. 1 mark: New menu option displayed in the correct sequence in menu, between 5 and Q. 1 mark: New selection statement added in an appropriate location within the loop in Main subroutine 1 mark: Selection statement has correct condition 1 mark: EnterUserData is called in an appropriate place 1 mark: EnterUserData is called with the parameter Individual. Max 5 if code not fully working Python <pre>def DisplayMenu(): print("Personal Fitness Application") print() print("MAIN MENU") print() print("1. Display User Information") print("2. Load Activity Log File") print("3. Add Activity") print("4. Display Activity Details") print("5. Sort Activities by Distance") print("6. Change Individual Person Data") print("Q. Quit") print() def Main(): Individual = PersonData("1001", "NameOne", "M", 21, 70.2) Choice = input("Enter D to use the default person data: ") if Choice != 'D': Individual = EnterUserData(Individual) ActivityLog=[] MenuOption = "0" while MenuOption != "Q": DisplayMenu() MenuOption = GetChoice() if MenuOption == "1": DisplayUserData(Individual) elif MenuOption == "2": ActivityLog = LoadActivity(Individual, ActivityLog) elif MenuOption == "3": ActivityLog = AddActivity(Individual, ActivityLog) elif MenuOption == "4": TypeOfActivity = input("What type of activity do you want to display: ") DisplayActivity(ActivityLog, TypeOfActivity, Individual) elif MenuOption == "5":</pre>	6 AO3=6

		<pre> ActivityLog = SortByDistance(ActivityLog) elif MenuOption == "6": Individual = EnterUserData(Individual) elif MenuOption == "Q": print("Goodbye, thank you for using this program") else: print("Invalid Menu Option") </pre> C# <pre> private static void DisplayMenu() { Console.WriteLine("Personal Fitness Application") Console.WriteLine(); Console.WriteLine("MAIN MENU"); Console.WriteLine(); Console.WriteLine("1. Display User Information"); Console.WriteLine("2. Load Activity Log File"); Console.WriteLine("3. Add Activity"); Console.WriteLine("4. Display Activity Details"); Console.WriteLine("5. Sort Activities by Distance"); Console.WriteLine("6. Change Individual Person Data"); Console.WriteLine("Q. Quit"); Console.WriteLine(); } static void Main(string[] args) { PersonData Individual; Individual.ID = "1001"; Individual.Name = "NameOne"; Individual.Gender = "M"; Individual.Age = 21; Individual.Weight = 70.2; Console.Write("Enter D to use the default person data:"); string Choice = Console.ReadLine(); if (Choice != "D") { Individual = EnterUserData(Individual); } Dim ActivityLog As New List(Of Exercise); string MenuOption = "0"; while (MenuOption != "Q") { DisplayMenu(); MenuOption = GetChoice(); if (MenuOption == "1") { DisplayUserData(Individual); } else if (MenuOption == "2") { ActivityLog = LoadActivity(Individual, ActivityLog); </pre>	
--	--	---	--

```

    }
    else if (MenuOption == "3")
    {
        ActivityLog = AddActivity(Individual, ActivityLog);
    }
    else if (MenuOption == "4")
    {
        Console.WriteLine("What type of activity do you want to
display: ");
        string TypeOfActivity = Console.ReadLine();
        DisplayActivity(ActivityLog, TypeOfActivity,
Individual);
    }
    else if (MenuOption == "5")
    {
        ActivityLog = SortByDistance(ActivityLog);
    }
    else if (MenuOption == "6")
    {
        Individual = EnterUserData(Individual);
    }
    else if (MenuOption == "Q")
    {
        Console.WriteLine("Goodbye, thank you for using this
application.");
    }
    else
    {
        Console.WriteLine("Invalid Menu Option");
    }
}
}

```

VB

```

Sub DisplayMenu()
    Console.WriteLine("Personal Fitness Application")
    Console.WriteLine()
    Console.WriteLine("MAIN MENU")
    Console.WriteLine()
    Console.WriteLine("1. Display User Information")
    Console.WriteLine("2. Load Activity Log File")
    Console.WriteLine("3. Add Activity")
    Console.WriteLine("4. Display Activity Details")
    Console.WriteLine("5. Sort Activities by Distance")
    Console.WriteLine("6. Change Individual Person Data")
    Console.WriteLine("Q. Quit")
    Console.WriteLine()
End Sub

Sub Main()
    Dim Individual As PersonData = New PersonData()
    Individual.ID = "1001"

```

		<pre> Individual.Name = "NameOne" Individual.Gender = "M" Individual.Age = 21 Individual.Weight = 70.2 Console.Write("Enter D to use the default person data: ") Dim Choice As String = Console.ReadLine() If Choice <> "D" Then Individual = EnterUserData(Individual) End If Dim ActivityLog As New List(Of Exercise) Dim MenuOption As String = "0" While MenuOption <> "Q" DisplayMenu() MenuOption = GetChoice() If MenuOption = "1" Then DisplayUserData(Individual) ElseIf MenuOption = "2" Then ActivityLog = LoadActivity(Individual, ActivityLog) ElseIf MenuOption = "3" Then ActivityLog = AddActivity(Individual, ActivityLog) ElseIf MenuOption = "4" Then Console.Write("What type of activity do you want to display: ") Dim TypeOfActivity As String = Console.ReadLine() DisplayActivity(ActivityLog, TypeOfActivity, Individual) ElseIf MenuOption = "5" Then ActivityLog = SortByDistance(ActivityLog) ElseIf MenuOption = "6" Then Individual = EnterUserData(Individual) ElseIf MenuOption = "Q" Then Console.WriteLine("Goodbye, thank you for using this application.") Else Console.WriteLine("Invalid Menu Option") End If End While End Sub </pre>	
--	--	---	--

Question	Part	Marking guidance	Total marks
07	2	1 mark: New menu option displayed and when option 6 is selected the user is requested to enter a name A. Incorrect menu option if matches Q7.1 Enter D to use the default person data: D Personal Fitness Application MAIN MENU 1. Display User Information 2. Load Activity Log File 3. Add Activity 4. Display Activities 5. Sort Activities by Distance 6. Change Individual Person Data Q. Quit Please enter your choice: 6 User Information: Enter person's name:	1 AO3=1

Question	Part	Marking guidance	Total marks
08	1	1 mark: New checks occur at appropriate locations. 1 mark: First two digits of time extracted. 1 mark: Convert first two digits to integer. 1 mark: Check integer value of first two digits is in range/out of range. 1 mark: Valid is set to false under the correct circumstance following their attempt at checking the first two digits. 1 mark: Last two digits of time extracted. 1 mark: Convert last two digits to integer. 1 mark: Check integer value of last two digits is in range/out of range. 1 mark: Valid is set to false under the correct circumstances following their attempt at checking the last two digits. 1 mark: No other part of the validation is affected by their attempt at the question and the invalid message is displayed under the correct circumstances. Alternative Marks for points 3, 4, 7 and 8 1 mark: Compare extracted substring with string boundary value. 1 mark: Comparison works as required to catch hours above boundary. 1 mark: Compare extracted substring with string boundary value. 1 mark: Comparison works as required to catch minutes above boundary. Python <pre>def EnterTime(): Valid = False while not Valid: Valid = True Time = input("Enter time of activity (hh:mm): ") if len(Time) != 5: Valid = False elif Time[2] != ":": Valid = False elif not Time[0:2].isdigit(): Valid = False elif not Time[3:5].isdigit(): Valid = False elif int(Time[0:2]) > 23: Valid = False elif int(Time[3:5]) > 59: Valid = False if not Valid: print("Time not in valid format") return Time</pre>	10 AO3=10

Alternative Solution

```

elif not Time[0:2].isdigit():
    Valid = False
elif not Time[3:5].isdigit():
    Valid = False
elif Time[0:2] > "23":
    Valid = False
elif Time[3:5] > "59":
    Valid = False
if not Valid:
    print("Time not in valid format")
return Time

```

C#

```

private static string EnterTime()
{
    string Time = "";
    bool Valid = false;
    int i = 0;
    while (!Valid)
    {
        Valid = true;
        Console.Write("Enter time of activity (hh:mm): ");
        Time = Console.ReadLine();
        if (Time.Length != 5)
        {
            Valid = false;
        }
        else if (Time[2] != ':')
        {
            Valid = false;
        }
        else if (!Int32.TryParse(Time.Substring(0, 2), out
i))
        {
            Valid = false;
        }
        else if (!Int32.TryParse(Time.Substring(3, 2), out
i))
        {
            Valid = false;
        }
        else if (Int32.Parse(Time.Substring(0, 2)) > 23)
        {
            Valid = false;
        }
        else if (Int32.Parse(Time.Substring(3, 2)) > 59)
        {
            Valid = false;
        }
    }
    if (!Valid)

```

		<pre> { Console.WriteLine("Time is not in a valid format."); } } return Time; } </pre> VB <pre> Function EnterTime() As String Dim Time As String = "" Dim Valid As Boolean = False Dim i As Integer While Not Valid Valid = True Console.Write("Enter time of activity (hh:mm): ") Time = Console.ReadLine() If Time.Length() <> 5 Then Valid = False ElseIf Time(2) <> ":" Then Valid = False ElseIf Not Integer.TryParse(Time.Substring(0, 2), i) Then Valid = False ElseIf Not Integer.TryParse(Time.SubString(3, 2), i) Then Valid = False ElseIf CInt(Time.Substring(0, 2)) > 23 Then Valid = False ElseIf CInt(Time.SubString(3, 2)) > 59 Then Valid = False End If If Not Valid Then Console.WriteLine("Time is not in a valid format.") End If End While Return Time End Function </pre>	
--	--	---	--

Question	Part	Marking guidance	Total marks
08	2	Screen capture(s) must match code from 08.1 1 mark: Both invalid times input and error message displayed followed by correct time which is accepted. <pre> MAIN MENU 1. Display User Information 2. Load Activity Log File 3. Add Activity 4. Display Activity Details 5. Sort Activities by Distance 6. Change Individual Person Data 7. Search Activities Q. Quit Please enter your choice: 3 Add an Activity ID: 1001 What day of week did you do the Activity (1 to 7): 2 What week of the year did you do the Activity (1 to 52): 3 Enter time of activity (hh:mm): 26:34 Time not in valid format Enter time of activity (hh:mm): 20:60 Time not in valid format Enter time of activity (hh:mm): 20:34 Enter average heart rate during activity (Beats per minute): </pre>	1 AO3=1

Section C

Question	Part	Marking guidance	Total marks
09	1	1. 1 mark: new selection structure added 2. 1 mark: selection structure distinguishes between cycling and other activities 3. 1 mark: condition for activity is case insensitive 4. 1 mark: correctly identifies category male cycling 5. 1 mark: correctly identifies category female cycling 6. 1 mark: correctly identifies category male non-cycling 7. 1 mark: correctly identifies category female non-cycling 8. 1 mark: attempt at both calculations using the four correct fields 9. 1 mark: correct calculation of calories burned when activity is cycling for females 10. 1 mark: correct calculation of calories burned when activity is cycling for males 11. 1 mark: no change to how calories burned for other activities are calculated following their attempt at adding a new selection structure to calculate calories burned when cycling. Max 10 if code not fully working Python <pre>def CalculateCalories(Activity, Individual): if Activity.Type.lower() == "cycling": if Individual.Gender == 'F': CaloriesBurned = Activity.Age * 0.10 - Activity.Weight * 0.07 + Activity.AverageHeartRate * 0.43 - 21 CaloriesBurned = CaloriesBurned * Activity.Duration / 4 elif Individual.Gender == 'M': CaloriesBurned = Activity.Age * 0.19 + Activity.Weight * 0.10 + Activity.AverageHeartRate * 0.62 - 56 CaloriesBurned = CaloriesBurned * Activity.Duration / 4 else: if Individual.Gender == 'F': CaloriesBurned = Activity.Age * 0.07 - Activity.Weight * 0.06 + Activity.AverageHeartRate * 0.45 - 20 CaloriesBurned = CaloriesBurned * Activity.Duration / 4 elif Individual.Gender == 'M': CaloriesBurned = Activity.Age * 0.20 + Activity.Weight * 0.09 + Activity.AverageHeartRate * 0.63 - 55 CaloriesBurned = CaloriesBurned * Activity.Duration / 4 return CaloriesBurned</pre> C# <pre>private static double CalculateCalories(Exercise Activity, PersonData Individual)</pre>	11 AO3=11

```

{
    double CaloriesBurned = 0;
    if (Activity.Type.ToLower() == "cycling")
    {
        if (Individual.Gender == "F")
        {
            CaloriesBurned = Activity.Age * 0.10 -
            Activity.Weight * 0.07 + Activity.AverageHeartRate * 0.43
            - 21;
            CaloriesBurned = CaloriesBurned * Activity.Duration
            / 4;
        }
        else if (Individual.Gender == "M")
        {
            CaloriesBurned = Activity.Age * 0.19 +
            Activity.Weight * 0.10 + Activity.AverageHeartRate * 0.62
            - 56;
            CaloriesBurned = CaloriesBurned * Activity.Duration
            / 4;
        }
    }
    else
    {
        if (Individual.Gender == "F")
        {
            CaloriesBurned = Activity.Age * 0.07 -
            Activity.Weight * 0.06 + Activity.AverageHeartRate * 0.45
            - 20;
            CaloriesBurned = CaloriesBurned * Activity.Duration
            / 4;
        }
        else if (Individual.Gender == "M")
        {
            CaloriesBurned = Activity.Age * 0.2 +
            Activity.Weight * 0.09 + Activity.AverageHeartRate * 0.63
            - 55;
            CaloriesBurned = CaloriesBurned * Activity.Duration
            / 4;
        }
    }
    return CaloriesBurned;
}

```

VB

```

Function CalculateCalories(Activity As Exercise,
    Individual As PersonData)
    Dim CaloriesBurned As Double
    If Activity.Type.ToLower() = "cycling" Then
        If Individual.Gender = "F" Then
            CaloriesBurned = Activity.Age * 0.10 -
            Activity.Weight * 0.07 + Activity.AverageHeartRate * 0.43
            - 21
            CaloriesBurned = CaloriesBurned * Activity.Duration
            / 4
        ElseIf Individual.Gender = "M" Then
            CaloriesBurned = Activity.Age * 0.19 +
            Activity.Weight * 0.10 + Activity.AverageHeartRate * 0.62
            - 56
            CaloriesBurned = CaloriesBurned * Activity.Duration
            / 4
        End If
    Else
        If Individual.Gender = "F" Then
            CaloriesBurned = Activity.Age * 0.07 -
            Activity.Weight * 0.06 + Activity.AverageHeartRate * 0.45

```

		<pre> - 20 CaloriesBurned = CaloriesBurned * Activity.Duration / 4 ElseIf Individual.Gender = "M" Then CaloriesBurned = Activity.Age * 0.2 + Activity.Weight * 0.09 + Activity.AverageHeartRate * 0.63 - 55 CaloriesBurned = CaloriesBurned * Activity.Duration / 4 End If End If Return CaloriesBurned End Function </pre>	
--	--	---	--

Question	Part	Marking guidance	Total marks																																																																						
09	2	Screen capture(s) must match code from Q9.1 1 mark: Activity input is in correct case (lower) and output for last row is displayed as shown, it may be the only row displayed. I. incorrect value for calories 1 mark: Correct display for the 6 Calories burned values with case as in output below. I. any additional cycling activities displayed. Please enter your choice: 4 What type of activity do you want to display: cycling Activity Data for ID: 1001 <table><tr><td>Activity</td><td>Calories Burned</td></tr><tr><td>Cycling</td><td>1016.80</td></tr><tr><td>Cycling</td><td>619.85</td></tr><tr><td>Cycling</td><td>548.80</td></tr><tr><td>Cycling</td><td>537.00</td></tr><tr><td>Cycling</td><td>370.10</td></tr><tr><td>cycling</td><td>1570.35</td></tr></table> Personal Fitness Application Note: The output above just shows the fourth and last column of the output, see full output expected below ID: 1001 <table><tr><th>Day No</th><th>Week No</th><th>Time</th><th>Activity</th><th>Duration (Minutes)</th><th>Distance (km)</th><th>Avg Heart Rate (BPM)</th><th>Calories Burned</th></tr><tr><td>4</td><td>3</td><td>18:30</td><td>Cycling</td><td>62</td><td>25.60</td><td>178</td><td>1016.80</td></tr><tr><td>5</td><td>3</td><td>13:30</td><td>Cycling</td><td>55</td><td>28.90</td><td>145</td><td>619.85</td></tr><tr><td>6</td><td>3</td><td>08:30</td><td>Cycling</td><td>70</td><td>31.00</td><td>123</td><td>548.80</td></tr><tr><td>7</td><td>3</td><td>07:30</td><td>Cycling</td><td>30</td><td>15.60</td><td>188</td><td>537.00</td></tr><tr><td>1</td><td>4</td><td>11:00</td><td>Cycling</td><td>20</td><td>12.80</td><td>192</td><td>370.10</td></tr><tr><td>2</td><td>4</td><td>08:10</td><td>cycling</td><td>87</td><td>33.30</td><td>189</td><td>1570.35</td></tr></table>	Activity	Calories Burned	Cycling	1016.80	Cycling	619.85	Cycling	548.80	Cycling	537.00	Cycling	370.10	cycling	1570.35	Day No	Week No	Time	Activity	Duration (Minutes)	Distance (km)	Avg Heart Rate (BPM)	Calories Burned	4	3	18:30	Cycling	62	25.60	178	1016.80	5	3	13:30	Cycling	55	28.90	145	619.85	6	3	08:30	Cycling	70	31.00	123	548.80	7	3	07:30	Cycling	30	15.60	188	537.00	1	4	11:00	Cycling	20	12.80	192	370.10	2	4	08:10	cycling	87	33.30	189	1570.35	2 AO3=2
Activity	Calories Burned																																																																								
Cycling	1016.80																																																																								
Cycling	619.85																																																																								
Cycling	548.80																																																																								
Cycling	537.00																																																																								
Cycling	370.10																																																																								
cycling	1570.35																																																																								
Day No	Week No	Time	Activity	Duration (Minutes)	Distance (km)	Avg Heart Rate (BPM)	Calories Burned																																																																		
4	3	18:30	Cycling	62	25.60	178	1016.80																																																																		
5	3	13:30	Cycling	55	28.90	145	619.85																																																																		
6	3	08:30	Cycling	70	31.00	123	548.80																																																																		
7	3	07:30	Cycling	30	15.60	188	537.00																																																																		
1	4	11:00	Cycling	20	12.80	192	370.10																																																																		
2	4	08:10	cycling	87	33.30	189	1570.35																																																																		

Question	Part	Marking guidance	Total marks
10	1	1. 1 mark: A subroutine with the correct name (SearchActivity) is created. 2. 1 mark: subroutine has only the correct parameter (ActivityLog). 3. 1 mark: Appropriate messages are displayed regarding distance and week number. A. Correct use of EnterDistance and EnterWeekNumber subroutines. 4. 1 mark: Value entered by user for distance and stored in appropriately named variable. 5. 1 mark: Value entered by user for week number and stored in appropriately named variable. 6. 1 mark: Loop that iterates once for every item in ActivityLog. 7. 1 mark: Loop in an appropriate place. 8. 1 mark: Selection statement within loop. 9. 1 mark: Selection statement has one correct condition. 10. 1 mark: Correct second condition linked with correct logical operator. 11. 1 mark: Output line contains correct 4 items of data. 12. 1 mark: Specified details for all activities that meet criteria are outputted. 13. 1 mark: If no activities that matched the criteria are found then appropriate message outputted once. <pre>def SearchActivity(ActivityLog): Found = False Distance = float(input("What distance do the activities have to be above (km): ")) WeekNo = int(input("What is the week number for the activities: ")) for Activity in ActivityLog: if Activity.WeekNo == WeekNo and Activity.Distance > Distance: Found = True print(Activity.DayNo, Activity.WeekNo, Activity.Type, Activity.Distance) if Found == False: print("No activities match the search criteria") #not this exact phrase print()</pre> C# <pre>private static void SearchActivity(List<Exercise> ActivityLog) { bool Found = false; Console.WriteLine("What distance do the activities have to be above (km): "); double Distance = Convert.ToDouble(Console.ReadLine()); Console.WriteLine("What is the week number for the activities: "); int WeekNo = Convert.ToInt32(Console.ReadLine()); foreach (Exercise Activity in ActivityLog) {</pre>	13 AO3=13

		<pre> if (Activity.WeekNo == WeekNo && Activity.Distance > Distance) { Found = true; Console.WriteLine(Activity.DayNo.ToString() + " " + Activity.WeekNo.ToString() + " " + Activity.Type + " " + Activity.Distance.ToString()); } } if (Found == false) { Console.WriteLine("No activities match the search criteria"); } Console.WriteLine(); } </pre> VB <pre> Sub SearchActivity(ByVal ActivityLog As List(Of Exercise)) Dim Found As Boolean = False Console.Write("What distance do the activities have to be above (km)") Dim Distance As Double = Console.ReadLine() Console.Write("What is the week number for the activities: ") Dim WeekNo as Integer = Console.ReadLine() For Each Activity In ActivityLog If Activity.WeekNo = WeekNo And Activity.Distance > Distance Then Found = True Console.WriteLine(Activity.DayNo.ToString() + " " + Activity.WeekNo.ToString() + " " + Activity.Type + " " + Activity.Distance.ToString()) End If Next If Found = False Then Console.WriteLine("No activities match the search criteria") End If Console.WriteLine() End Sub </pre>	
--	--	---	--

Question	Part	Marking guidance	Total marks
10	2	1 mark: DisplayMenu subroutine contains an appropriate message for the new menu option. 1 mark: Main subroutine contains a new selection statement in the WHILE loop with a condition that matches the new menu option in DisplayMenu. 1 mark: SearchActivity subroutine with correct parameter called in selection structure. <pre> def DisplayMenu(): print("Personal Fitness Application") print() print("MAIN MENU") print() print("1. Display User Information") print("2. Load Activity Log File") print("3. Add Activity") print("4. Display Activity Details") print("5. Sort Activities by Distance") print("6. Change Individual Person Data") print("7. Search Activities") print("Q. Quit") print() def Main(): Individual = PersonData() Choice = input("Enter D to use the default person data: ") if Choice != 'D': Individual = EnterUserData(Individual) ActivityLog=[] MenuOption = "0" while MenuOption != "Q": DisplayMenu() MenuOption = GetChoice() if MenuOption == "1": DisplayUserData(Individual) elif MenuOption == "2": ActivityLog = LoadActivity(Individual, ActivityLog) elif MenuOption == "3": ActivityLog = AddActivity(Individual, ActivityLog) elif MenuOption == "4": TypeOfActivity = input("What type of activity do you want to display: ") DisplayActivity(ActivityLog, TypeOfActivity, Individual) elif MenuOption == "5": ActivityLog = SortByDistance(ActivityLog) elif MenuOption == "6": Individual = EnterUserData(Individual) elif MenuOption == '7': SearchActivity(ActivityLog) elif MenuOption == "Q": print("Goodbye, thank you for using this program") else: print("Invalid Menu Option") </pre>	3 AO3=3

		C# <pre> private static void DisplayMenu() { Console.WriteLine("Personal Fitness Application"); Console.WriteLine(); Console.WriteLine("MAIN MENU"); Console.WriteLine(); Console.WriteLine("1. Display User Information"); Console.WriteLine("2. Load Activity Log File"); Console.WriteLine("3. Add Activity"); Console.WriteLine("4. Display Activity Details"); Console.WriteLine("5. Sort Activities by Distance"); Console.WriteLine("6. Change Individual Person Data"); Console.WriteLine("7. Search Activities"); Console.WriteLine("Q. Quit"); Console.WriteLine(); } static void Main(string[] args) { PersonData Individual; Individual.ID = "1001"; Individual.Name = "NameOne"; Individual.Gender = "M"; Individual.Age = 21; Individual.Weight = 70.2; Console.Write("Enter D to use the default person data: "); string Choice = Console.ReadLine(); if (Choice != "D") { Individual = EnterUserData(Individual); } List<Exercise> ActivityLog = new List<Exercise>(); string MenuOption = "0"; while (MenuOption != "Q") { DisplayMenu(); MenuOption = GetChoice(); if (MenuOption == "1") { DisplayUserData(Individual); } else if (MenuOption == "2") { ActivityLog = LoadActivity(Individual, ActivityLog); } else if (MenuOption == "3") { ActivityLog = AddActivity(Individual, ActivityLog); } else if (MenuOption == "4") { Console.Write("What type of activity do you want to display: "); string TypeOfActivity = Console.ReadLine(); DisplayActivity(ActivityLog, TypeOfActivity, Individual); } else if (MenuOption == "5") { ActivityLog = SortByDistance(ActivityLog); } else if (MenuOption == "6") { Individual = EnterUserData(Individual); </pre>	
--	--	---	--

```

    }
    else if (MenuOption == "7")
    {
        SearchActivity(ActivityLog);
    }
    else if (MenuOption == "Q")
    {
        Console.WriteLine("Goodbye, thank you for using this
application.");
    }
    else
    {
        Console.WriteLine("Invalid Menu Option");
    }
}
}

```

VB

```

Sub DisplayMenu()
    Console.WriteLine("Personal Fitness Application")
    Console.WriteLine()
    Console.WriteLine("MAIN MENU")
    Console.WriteLine()
    Console.WriteLine("1. Display User Information")
    Console.WriteLine("2. Load Activity Log File")
    Console.WriteLine("3. Add Activity")
    Console.WriteLine("4. Display Activity Details")
    Console.WriteLine("5. Sort Activities by Distance")
    Console.WriteLine("6. Change Individual Person Data")
    Console.WriteLine("7. Search Activities")
    Console.WriteLine("Q. Quit")
    Console.WriteLine()
End Sub

Sub Main()
    Dim Individual As PersonData = New PersonData()
    Individual.ID = "1001"
    Individual.Name = "NameOne"
    Individual.Gender = "M"
    Individual.Age = 21
    Individual.Weight = 70.2
    Console.Write("Enter D to use the default person data:
")
    Dim Choice As String = Console.ReadLine()
    If Choice <> "D" Then
        Individual = EnterUserData(Individual)
    End If
    Dim ActivityLog As New List(Of Exercise)
    Dim MenuOption As String = "0"
    While MenuOption <> "Q"
        DisplayMenu()
        MenuOption = GetChoice()
        If MenuOption = "1" Then
            DisplayUserData(Individual)
        ElseIf MenuOption = "2" Then
            ActivityLog = LoadActivity(Individual, ActivityLog)
        ElseIf MenuOption = "3" Then
            ActivityLog = AddActivity(Individual, ActivityLog)
        ElseIf MenuOption = "4" Then
            Console.Write("What type of activity do you want to
display: ")
            Dim TypeOfActivity As String = Console.ReadLine()
            DisplayActivity(ActivityLog, TypeOfActivity,
Individual)
        ElseIf MenuOption = "5" Then
            ActivityLog = SortByDistance(ActivityLog)

```

		<pre> ElseIf MenuOption = "6" Then Individual = EnterUserData(Individual) ElseIf MenuOption = "7" Then SearchActivity(ActivityLog) ElseIf MenuOption = "Q" Then Console.WriteLine("Goodbye, thank you for using this application.") Else Console.WriteLine("Invalid Menu Option") End If End While End Sub </pre>	
--	--	---	--

Question	Part	Marking guidance	Total marks
10	3	Screen capture(s) must match code from Q10.1 and Q10.2 1 mark: Appropriate prompts for distance and week number. 1 mark: Correct output is displayed. A. Each required field output on a separate line I. Any additional activities displayed which meet the correct criteria. MAIN MENU <pre> 1. Display User Information 2. Load Activity Log File 3. Add Activity 4. Display Activities 5. Sort Activities by Distance 6. Change Individual Person Data 7. Search Activities Q. Quit Please enter your choice: 7 What distance do the activities have to be above (km): 10 What is the week number for the activities: 3 2 3 Running 10.9 2 3 Running 11.0 3 3 Running 11.2 4 3 Running 12.8 4 3 Cycling 25.6 5 3 Cycling 28.9 6 3 Cycling 31.0 7 3 Cycling 15.6 Personal Fitness Application </pre>	2 AO3=2