L3 Lead Examiner Report 1806

BTEC

Summer 2018

Level 3 National in Computing

Unit 4: Software Design and Development Project (31771H)

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 websites at www.edexcel.com or www.btec.co.uk for our BTEC qualifications.

Alternatively, you can get in touch with us using the details on our contact us page at www.edexcel.com/contactus.

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: www.edexcel.com/teachingservices.

You can also use our online Ask the Expert service at www.edexcel.com/ask. You will need an Edexcel username and password to access this service.

earson: 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 educatiPon 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

August 2018
Publications Code 31771_1806_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, 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 should be 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 test, because then it would not take into account that a test might be slightly easier or more difficult than any other.

Grade boundaries for this, and all other papers, are on the website via this link:
http://www.edexcel.com/iwantto/Pages/grade-boundaries.aspx

Unit 4: Software Design and Development Project (31771H)

Grade	**Unclassified**	**Level 3**		
		P	**M**	**D**
Boundary Mark	**0**	**24**	**37**	**50**

Introduction

This was the first examination season for Level 3 BTEC Computing Unit 3: 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 good although we have do not have a comparable series as this is the first sitting of this unit. However, it was evident that learners were well prepared for the rigour of this assessment.

The performance on Activity 1 was as expected with many learners picking up marks in band 2. Most of the responses used BCS symbols and learners made a good attempt at breaking down the requirements into relevant parts. The number of blank responses was also significantly low.

Activity 2 was of a high standard and demonstrated the learner ability to apply pseudocode design methodologies to a scenario. Some learners produced pseudocode which was too close to the coding which resulted in some marks being lost.

Activity 3 & 4 (testing) was particularly poor and resulted in most learners only accessing mark band 1. It is recommended that centres reinforce what a test plan consists of and the importance of testing throughout the whole process. In most cases, the testing carried out did not show evidence of any errors encountered which is essential for accessing higher marks.

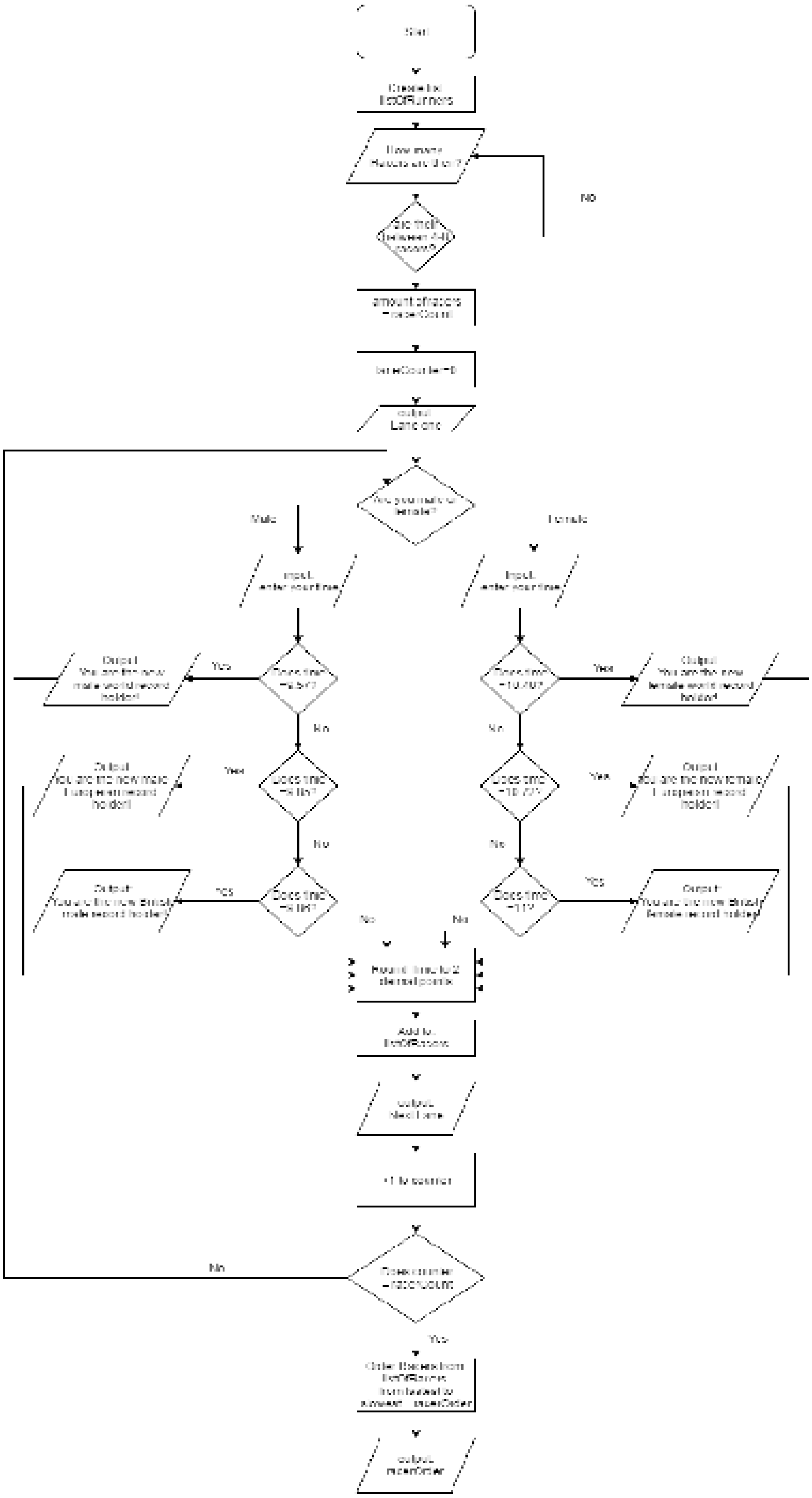
Activity 4 (Coding) was carried out to a good standard by the learners. Some learners were awarded full marks as they produced a working solution along with detailed comments.

Finally, the evaluations (activity 5) were of a good standard and most of the learners' accessed bands two and three. Some learners only produced a review 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.

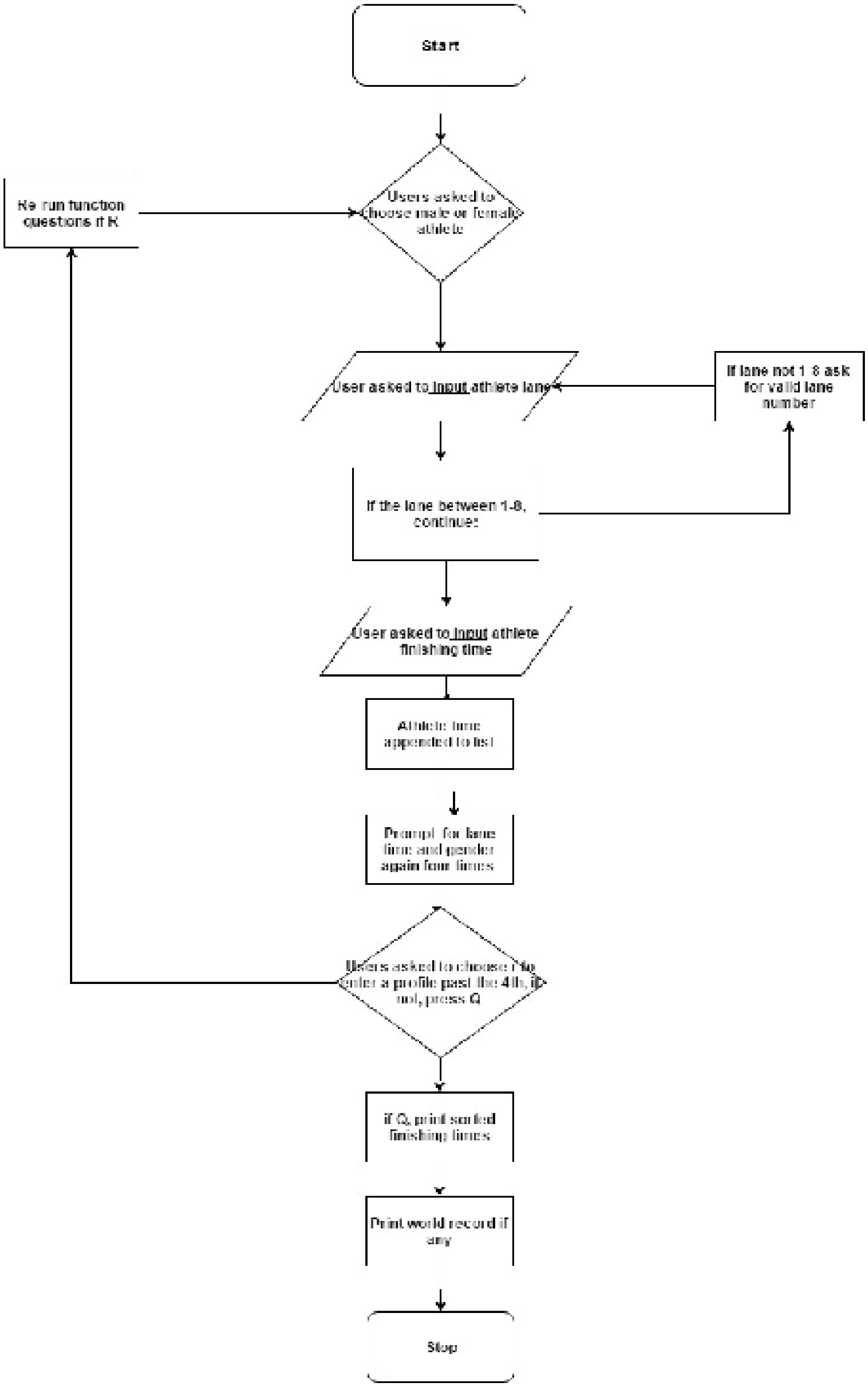
Activity 1


Start
Create list listOfRunners
How many Racers are there?
Are there between 2-8 racers?
No
amount of racers = raceCount
laneCounter=0
output Lane one
Are you male or female?
Male
Female
Input: enter your time
Input: enter your time
Does time <9.58?
Yes
Output You are the new male world record holder!
No
Does time <9.86?
Yes
Output You are the new male European record holder!
No
Does time <9.87?
Yes
Output You are the new British male record holder!
No
Does time <10.49?
Yes
Output You are the new female world record holder!
No
Does time <10.73?
Yes
Output You are the new female European record holder!
No
Does time <11?
Yes
Output You are the new British female record holder!
No
Round time to 2 decimal points
Add to listOfRacers
output Next Lane
+1 to counter
Does counter = raceCount
No
Yes
Order Racers from listOfRacers from fastest to slowest = raceOrder
output raceOrder

The learner has had a good go at the flow chart for the specified problem.

British Computer Society (BCS) flowchart symbols have been used accurately throughout as well as the breaking down of requirements into component parts that are detailed and relevant.

The flowchart shows full coverage of inputs, outputs and processes using naming conventions appropriate to the scenario consistently. Although there is no validation checking on male or female and variable names are inconsistent it is still deserving of a mark in **band 3 (8 marks).**


Start
Re run function questions if R
Users asked to choose male or female athlete
User asked to input athlete lane
If lane not 1-8 ask for valid lane number
If the lane between 1-8, continue
User asked to input athlete finishing time
Athlete time appended to list
Prompt for lane time and gender again four times
Users asked to choose r to enter a profile past the 4th, if not, press Q
if Q, print sorted finishing times
Print world record if any
Stop

The learner has addressed all aspects of the flow chart for the specified problem.

British Computer Society (BCS) flowchart symbols have been used but some are inaccurate. There is some evidence of breaking down of requirements into component parts that are relevant.

The flowchart shows some coverage of inputs, outputs and processes using naming conventions, but it does not meet minimum runner requirements.

Links between component parts are incomplete with limited procedures for handling unexpected events.

Mark in **band 1 (3 marks)**.

Activity 2

```
runnersList = []
runnersTime = None
runnersGender = None
runnersLane = None
define gendCheck
        global runnersGender
        print: This program requires you to input a minimum of 4 athlete profiles
        runnersGender = input: Please input m for a male and f for a female
        runnersGender = runnersGender.lower()
define laneCheck
        global runnersLane
        while True:
                string runnersLane = input: Please choose the athlete's lane (1-8)
                try:
                runnersLane = integer (string runnersLane)
                except ValueError:
                print: Please enter a valid number 1 through 8
        else:
                if runnersLane more than 0 and runnersLane less than 9:
                print: Lane recorded
                break out of loop here
        else:
                print: Please input a lane number between 1 and 8
define timeCheck
        global runnersTime
        runnersTime = input: Enter the athlete's finishing time(9.58 for example)
        runnersList added to list as float with .append
define runFunc
        call gendCheck
        call laneCheck
        call timeCheck
define runAgain
        global runnersList
        while True:
                rerun = input: Please input r to input more profiles or q to quit
                if rerun is “r”
                call gendCheck
                call laneCheck
                call timeCheck
                elif rerun is “q”
                runnersList is sorted (.sort)
                print runnersList
                call worldRecord
                break out of loop here
def runInitial
        call runFunc
        call runFunc
```

```
    call runFunc
    call runFunc
    call runAgain
def worldRecord
    global runnersTime
    global runnersGender
    if runnersTime less than "9.57" and runnersGender is "m":
        print Congratulations! Male 100m World Record!
    elif runnersTime less than "9.86" and runnersGender "m":
        print Congratulations! Male 100m European Record!
    elif runnersTime less than "9.87" and runnersGender "m":
        print Congratulations! Male 100m British Record!
    elif runnersTime less than "10.49" and runnersGender "f":
        print Congratulations! Female 100m World Record!
    elif runnersTime less than "10.73" and runnersGender "f":
        print Congratulations! Female 100m European Record!
    elif runnersTime less than "10.99" and runnersGender "f":
        print Congratulations! Female 100m British Record!
    else:
        print No record broken
call runInitial
```

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. Appropriate naming conventions have been used and precise use of logical operations. The solution is too close to Python code, need more readable structured English to obtain full marks.

Mark in **band 3 (8 marks)**.

```
 START
From decimal import decimal
INITIALISE m_w= 9.58
INITIALISE m_e= 9.86
INITIALISE m_b= 9.87
INITIALISE w_w= 10.49
INITIALISE w_e= 10.73
INITIALISE w_b= 10.99
Display '’the total number of athletes participating in this race’’
        'IF the number of participants ≥4
                'IF the number of participants ≤8
                        WHILE COUN=1
                                READ '’Enter the finishing time for the participants’’
        End loop
        Else
                Display error
CALCULATE the finishing time from fastest to slowest
IF a participant male
        IF the answer < m_w
        Display '’you have achieved the world record’’
Else IF the answer < m_e
        Display '’you have broken the European Record’’
Else IF the answer < m_b
        Display '’you have broken the world record’’
Else Display '’No Record Was Broken’’
        IF a participant female
IF the answer < w_w
        Display '’you have achieved the world record’’
Else IF the answer < w_e
        Display '’you have broken the European Record’’
Else IF the answer < w_b
        Display '’you have broken the world record’’
Else Display '’No Record Was Broken’’
        Display the finishing time
STOP
```

The learner has produced a structure which shows appropriate and consistent use of hierarchy and indentation, providing clarity and mostly readable pseudocode. However, the pseudocode will not provide a working solution. The Loop for participants times check is not logically correct and there is no logic for sorting the finishing times.

Mark in **band 1 (3 marks)**.

Activity 3

Test Plan

I Used Python Test Number	**Purpose of test**	**Test Data**	**Expected Result**
1	If I enter a data type that isn't an integer will it reject that input?	q	Because I have used try except else, it should reject any other data type than integer
2	If I enter a number lower than 4 when entering how many racers are in the race, will it deny me?	3	It should fail and fall into the else statement telling me to try again
3	If I enter a number higher than 8 when entering how many racers are in the race, will it deny me?	9	It should fail and fall into the else statement telling me to try again
4	If I enter a number that fits in between 4-8 will the program continue?	6	It Should allow this number to continue the code
5	If I enter the number 4 will it continue the code or deny me	4	As I entered greater than 3 in the code it means that 3 doesn't count as seen before but 4 should
6	If I enter the number 8 will it continue the code or deny me	8	As I entered less than 9 in the code it means that 9 doesn't
7	If I enter anything other than a letter in my code will it reject it	1	Because I have used the function .isalpha() which only allows the input of numbers from a-z, If I enter e number it should loop round until I enter only letters
8	If I enter only letters for my name, will the code move on to the next loop	Joe	As mentioned before .isalpha only accepts a-z in both caps so it should allow the program to continue
9	Will it add on my position after I input my name	Joe	Because python counts 1 as 0, to see if this has worked my position needs to be one, this should work as when a valid name is entered, the variable position is changed to add +1 to itself every time the code loops round

10	If I enter a gender that is not Female or Male will the code reject it?	N	As I have set it so that if gender is equal to M for male or F for female, Having anything else
11	If I Enter M will the code move on to the next loop?	M	As previously mentioned, I have set it so that the only way it will leave that gender loop is if the input is equal to M or F so it shall continue the code
12	If I Enter F will the code move on to the next loop?	F	As previously mentioned, I have set it so that the only way it will leave that gender loop is if the input is equal to M or F so it shall continue the code
13	If I enter a lower case M will it still accept it?	m	Because I used the function .upper and had the variable(gender) equal .upper, this means it should accept both upper and lower case
14	If I enter a lower case F will it still accept it?	f	Because I used the function .upper and had the variable(gender) equal .upper, this means it should
15	If I enter a letter for the variable time will it reject me?	q	Because I have used try except else it should deny the use of a string when it is set to only accept floats
16	If I enter a valid float for the variable time will it be taken?	16.34	As try except else has set it to floats this should go into the else as it is not a value error
17	If I enter a time below 9.58 for male Will it be recognised as a world record?	8.67	As I have said that if time less than 9.58 meaning it should accept all numbers lower than 9.57 as world records
18	If I enter exactly 9.57 for male will it state that I am the world record holder?	9.57	As 9.57 is below 9.58 it should state I have broken the world record
19	If I enter exactly 9.58 for male will it state that I am the European record holder or the world record holder?	9.58	As to break a record you need to be underneath it this should tell me that I have broken the record in Europe as 9.58 isn't less than 9.58 but equal to
20	If I enter exactly 9.87 for male will it state that I am the British record holder or the European record holder?	9.87	As to break a record you need to be underneath it this should tell me that I have broken the record in Brittan as 9.87 isn't less than 9.87 but equal to
21	If I enter above 9.88 for male will it not award me with a record	9.88	As it is set for anything equal to or above 9.88 which is the last possible record to get so it should just go to the else statement
22	If I enter a time below 10.49 for female Will it be recognised as a world record?	10.35	As I have said that if time less than 10.50 meaning it should accept all numbers lower than 10.49 as world records
23	If I enter exactly 10.49 for female will be	10.49	As 10.49 is below 10.50 it should state I have broken the world record

	accepted as a new world record		
24	If I enter exactly 10.50 for female will it state that I am the European record holder or the world record holder?	10.50	As to break a record you need to be underneath it this should tell me that I have broken the record in Europe as 10.50 isn’t less than 10.50 but equal to
5	If I enter exactly 10.73 for female will it state that I am the British record holder or the European record holder?	10.73	As to break a record you need to be underneath it this should tell me that I have broken the record in Brittan as 10.73 isn’t less than 10.73 but equal to
26	If I enter above 10.99 will I be awarded a record?	11	As it is set for anything equal to or above 11 which is the last possible record to get so it should just go to the else statement
27	After finishing inputting all of the time will the code loop back round again the set amount of times	4	Dependent on how many runners I select should be how many time the code loops in the for statement so it should loop round 4 times
28	After looping around 4 times will the code output all of the information appended to the list	4	By using the function .append() it means that after a loop all of the information that I put in the append function will go into the table.
29	Will all of the information be ordered in the print	4	By using the function.sort() and by placing time(the variable that needs to be sorted) as the first thing to be appended into the list, means that it should be able to sort the times from fastest to slowest.

The learner has produced a thorough test plan which includes a range of data. Expected results are specific and accurate based in the test data.

Mark in **band 3 (6 marks)**.

Test number	Purpose of test	Test data	Expected result
1	The program accepts the gender male	I will test to see if my code accepts me inputting the gender male for the program	The program will accept the selection and continue with the program
2	The program accepts the gender female	I will test to see if my code accepts me inputting the gender female for the program	The program will accept the selection and continue with the program
3	The program doesn't accept letters in the score	I will test to see if the text box for the score will accept letters	When I try to run the program and it will stop and display a message
4	The program doesn't let me enter a number without two decimal places	I will test if the text box for the score allows me to enter a two decimal places	The program will accept the numbers and continue
5	The program doesn't let me enter more than 2 decimal places	I will test to see if the program lets me enter a number into the score with more than 2 decimal places	The program will stop me from entering more than 2 decimal places
6	The program exits when the exit button is selected	I will select the exit button and see if the program is exited	The program will be exited
7	The program clears all text boxes when the	I will test to see If all text boxes are cleared	All text boxes will be cleared of any text

The learner has produced a test plan that is too narrow to confirm a working solution as no test data is provided. Expected results are generic based on identified test data.

Mark in **band 1 (2 marks)**.

Activity 4 - Program

PYTHON

```
------------------------------------------------------------------------------------------
--------------------------------------------
#This is the list in the code that all of the information will be appended to
listOfRunners = []
#These are the two counters that continuosuly add when going throught the for
loop or when the variable is changed
#in the case of positoin
position = 0
counter = 0
#Their are many of these while loops in the code which will keep going throught
the loop until is breaks
while True:
try:
print("How many racers are thier (4-8)")
racerNumber = int(input())
#This is the try except else validation which means instead of the code crashing
when given the wrong data type
#it loops back around to the start with an error message telling the user to try
again
#When the user enters a valid ansuer it falls under the else statment and the
loop breaks
except ValueError:
print("Please enter a valid number")
else:
if racerNumber >3 and racerNumber <9:
#This is move validation which is neede after a valid intiger has been inputed,
this will check to seen if the
#number is greater than 3 or less than 9 meaning any number between 4-8 will
break the loop but anything else
#will enter the else statment and have the user re-enter a number
break
else:
print("Please enter a number between 4-8")
#This is the for loop, for loops only loop around the code a set number of times,
in this case ive set it so that
#it needs to go from 0 - the number that the user enterd, meaning that if the
user entered 5 it will loop around
#5 times
for counter in range(0,racerNumber):
while True:
print("Enter your name")
name = input()
#the function .isalpha() is used to only accept data between a-z lower and upper
case, if the data is accepted
#then plus 1 will be added to the position counter which will show the user what
lane the athlete is in and because
#position = 1 the first athlete will be 0 + 1 = 1 so will be in lane 1
if name.isalpha():
```

```
position = position + 1
print(name, "is in position", position)
break
else:
print("Please enter a valid name")
while True:
print("Are you male of female (M/F)?")
gender = input()
#.upper() is used so that the print statment can accept both lower case and
upper case as the same and will be
# printed in lower case
gender = gender.upper()
#This if statment means that if the user did not enter either lower or upper case
M of F(because of .upper())
#The loop will repeat and ask for another input
if gender == "M" or gender == "F":
break
else:
print("please enter either male or female")
while True:
#Here we see that if then gender was M then the user will be put into this loop
if gender == "M":
try:
print("Enter your time in the 100m sprint(seconds)")
time = float(input())
except ValueError:
print("Please enter a valid number")
else:
#When the user enters a valid float thier time will enter these if, elif and else
statments, this is to find out
#if the athlete has broken a world record, it does this by using less than or
greater than staments so their time
#can be put in a ceitan group, wolrd record, european record, british record and
no record
if time <9.58:
print("You are the new male world record holder!")
listOfRunners.append([time, position, gender, name])
#The append function then puts all of the information entered in the first loop
into the list and will then
#repeat itself the remeaning number of times
break
elif time >9.57 and time <9.87:
print("You are the new male Europeian record holder!")
listOfRunners.append([time, position, gender, name])
break
elif time >9.86 and time <9.88:
print("You are the new male British record holder!")
listOfRunners.append([time, position, gender, name])
break
else:
listOfRunners.append([time, position, gender, name])
break
if gender == "F":
```

```
#Here we see that if then gender was F then the user will be put into this loop
try:
print("Enter your time in the 100m sprint(seconds)")
time = float(input())
except ValueError:
print("Please enter a valid number")
else:
if time <10.50:
print("You are the new female world record holder!")
listOfRunners.append([time, position, gender, name])
break
elif time >10.49 and time <10.73:
print("You are the new female Europeian record holder!")
listOfRunners.append([time, position, gender, name])
break
elif time >10.72 and time <11:
print("You are the new female British record holder!")
listOfRunners.append([time, position, gender, name])
break
else:
listOfRunners.append([time, position, gender, name])
#The two genders have been put into different loops as they have different records times to one another
break
listOfRunners.sort()
#This function here will sort out the first thing appended into the table which is time, the one thing that
#needs to be sorted, it will print the times from fastest to slowest
for i in listOfRunners:
#This for statment means it will repeat the print stament for all of the inputted data as if their was no
#for loop the . format functoin would only be able to format the first loop of data
runnerFormat = (",Name: {3}\n ,Position: {1}\n ,Gender: {2}\n ,Time: {0:.2f}\n".format(i[0], i[1], i[2], i[3]))
#the {0:.2f} is used so that it will round the data at position 0(time) to 2 decimal points
print(runnerFormat)
```

The learner has produced a program that fully meets all the requirements. Improvements could be made to the code such as use of functions instead of nested ELIF but it is still efficient. Accurate syntax and indentation has 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 robust program being created.

Mark in **band 4 (24 marks)**.

```
 # Program that allows the user to input a minimum of 4 profiles for an athlete's
time, gender and lane
# Allows the user to choose gender
# Allows the user to input times in lane order (see comment under laneCheck)
# Outputs finishing times fastest to slowest in the list
# Outputs a world record if any athlete achieved one
# Defined variables that will be used throughout the functions
runnersList = []
runnersTime = None
runnersGender = None
runnersLane = None
# Function to check for an athlete's gender
def gendCheck():
global runnersGender
print("This program requires you to input a minimum of 4 athlete profiles")
runnersGender = input("Please input m for a male and f for a female")
runnersGender = runnersGender.lower()
# Function to check for an athlete's lane
def laneCheck():
global runnersLane
while True:
strrunnersLane = input("Please choose the athlete's lane (1-8)") # Allows the
user to choose whichever lane order they want (not 1 through 8 linearly)
try:
runnersLane = int(strrunnersLane)
except ValueError:
print("Please enter a valid number 1 through 8")
else:
if runnersLane > 0 and runnersLane < 9:
print("Lane recorded")
break
else:
print("Please input a lane number between 1 and 8")
# Function to check for an athlete's time and add it to the list
def timeCheck():
global runnersTime
runnersTime = input("Enter the athlete's finishing time(9.58 for example)")
runnersList.append(float(runnersTime))
# Function to allow the above three functions to run when it's called
def runFunc():
gendCheck()
laneCheck()
timeCheck()
# Function to allow the user a minimum of 4 profiles with a choice to continue or
abort and print the list
def runAgain():
global runnersList
while True:
rerun = input("Please input r to input more profiles or q to quit")
if rerun == "r":
gendCheck()
laneCheck()
```

```
timeCheck()
elif rerun == "q":
runnersList.sort()
print(runnersList)
worldRecord() # World record function is called here
break
# Function that makes runFunc run 4 times before giving the option to abort the
program
def runInitial():
runFunc()
runFunc()
runFunc()
runFunc()
runAgain()
# Function to print whether or not an athlete achieved a world record
def worldRecord():
global runnersTime
global runnersGender
if runnersTime < "9.57" and runnersGender == "m":
print("Congratulations! Male 100m World Record!")
elif runnersTime < "9.86" and runnersGender == "m":
print("Congratulations! Male 100m European Record!")
elif runnersTime < "9.87" and runnersGender == "m":
print("Congratulations! Male 100m British Record!")
elif runnersTime < "10.49" and runnersGender == "f":
print("Congratulations! Female 100m World Record!")
elif runnersTime < "10.73" and runnersGender == "f":
print("Congratulations! Female 100m European Record!")
elif runnersTime < "10.99" and runnersGender == "f":
print("Congratulations! Female 100m British Record!")
else:
print("No record broken")
# Function call to allow the whole program to run
runInitial()
```

The learner has produced a program that meets some of the requirements. Some accurate validation used and the program minimises one common error. Outputs are accurate and mostly informative. Accurate syntax and indentation used along with some logical structure. Commenting of the code is not very detailed and a third party would have difficulty with it.

Mark in **band 2 (10 marks)**.

Activity 4 – Testing

Test number	Purpose of test	Test data	Expected result	Actual result	Comments
1	The program accepts the gender male	I will test to see if my code accepts me inputting the gender male for the program	The program will accept the selection and continue with the program	The program does accept the input male into the gender text box	Whilst it accepted the input I do not know whether it stores the input or not
2	The program accepts the gender female	I will test to see if my code accepts me inputting the gender female for the program	The program will accept the selection and continue with the program	The does accept the gender input as female into the gender textbox	Whilst it accepted the input I do not know whether it stores the input or not
3	The program doesn’t accept letters in the score	I will test to see if the text box for the score will accept letters	When I try to run the program and it will stop and display a message	This didn’t work as expected this is due to the code being associated with the text box via click event rather than when someone types	I will attempt to fix this via changing the code to be associated with the user input rather than the click event of the score text box
4	The program doesn’t let me enter a number without two decimal places	I will test if the text box for the score allows me to enter a two decimal places	The program will accept the numbers and continue	This didn’t work as expected the code accepts user input that isn’t to two decimal places	I do not know of any code that will prevent this from happening. It have to rely on the user inputting two decimal places. However, I

					will still attempt to find a solution
5	The program doesn’t let me enter more than 2 decimal places	I will test to see if the program lets me enter a number into the score with more than 2 decimal places	The program will stop me from entering more than 2 decimal places	This didn’t work as expected the code accepts user input that isn’t over two decimal places	I do not know of any code that will prevent this from happening. It have to rely on the user inputting two decimal places. However, I will still attempt to find a solution
6	The program exits when the exit button is selected	I will select the exit button and see if the program is exited	The program will be exited	The program exited once I pressed the exit button as expected	This was done in an efficient manner
7	The program clears all text boxes when the clear button is selected	I will test to see If all text boxes are cleared of text when the clear button is selected	All text boxes will be cleared of any text	The program cleared all text boxes of text as expected	This was done in an efficient manner
8	The program doesn’t allow me to enter a negative number	I will test to see if the program allows me to enter a negative number by entering a negative number and continuing the program	It will stop me and display a message stating “Please enter a number below 50 and above 1”	The program does allow me to enter a negative number this again is due to the code being associated with the click event rather than the	I will attempt to fix this error via changing the code to be associated with the users input rather than the click event of the text box

				users input	
9	The program doesn't allow me to enter a number above 50	I will test to see if the program allows me to enter a number above 50	It will stop me and display a message stating "Please enter a number below 50 and above 1"	The program does allow me to enter a number over 50 this again is due to the code being associated with the click event rather than the users input	I will attempt to fix this error via changing the code to be associated with the users input rather than the click event of the text box
10	Seeing if it Stores gender	I will test to see if my program stores the gender of the user into the array	The program will store the gender into the array successfully	The program didn't work as expected as the program didn't store the users input as the variable	I will attempt to fix this problem via connecting the users input from gender to the gender variable
11	Does the program store the score	I will test to see if my program stores the data of the score into the array	The program will store the score into the array successfully	The program didn't work as expected as it didn't store the score as a variable	I will attempt to fix this via making the score variable be the input from the user from the score text box
12	The display scores button	I will test to see if the program displays all arrays when selecting the display scores button	The full array will be shown successfully with no errors	This didn't work as expected as the array was not shown	I will attempt to fix this via making the array store all data added to it, group and sort the

					data correctly
13	The number of athletes in the array	I will test to see If the program allows me to display the array when less than 4 athletes has been stored	The program will not allow me to proceed and appear with a message	The program works as expected and doesn’t allow me to continue if the people in the array is lower than 4	This was done in an efficient manner
14	The number of athletes in the array	I will test to see if the program allows me to display the array when more than 8 athletes have been stored	The program will not allow me to continue and will display a message stating that I cannot have more than 8 athletes stored	The program stopped me from progressing and displayed the message “you cannot store more than 8 athletes” as predicated	This was done in an efficient manner
15	Meeting the requirements to have a championship displayed	I will test to see if the program will display a message to me if someone achieves a championship record	The program will appear with a message stating “someone has won a *championship type* record”	The program didn’t work as expected as gender was not declared as a string and any code to connect two statements together is unknown to me	I will attempt to find a connecter to connect the two requirements for gaining a championship (gender and score) and declare gender as a string as to accept male or female as an input
16	Move event	I will test to see if text	Text will appear	The program	I will attempt to

		appears when the mouse is moved over the add to score button	explaining what the button does	didn’t work as expected this is due to the text not being associated as a string	fix this problem via changing the label and text to be displayed as a string
17	Spelling punctuation and grammar	In this I will be testing to see if all spelling and punctuation is correct as well as to check to see if grammar has been used correctly	There will be no spelling punctuation or grammar mistakes	There were no spelling punctuation or grammar mistakes that I could find	No spelling punctuation or grammar mistakes where found however, it’s possible there was a mistake that I didn’t notice

The learner has not identified any errors, comments do not demonstrate an understanding of the testing process. Evidence of some errors being resolved is required for higher mark bands.

Mark in **band 1 (1 mark)**.

Document for Activities 3 and 4

Test Plan (add additional rows as required)

No	Purpose Data	Test Data	Expected Result	Actual Result	Comment
1	To check the number of athlete participating is valid or not	6	The expected result would be computer accepts the data and brings up the next question	The program worked successfully when I typed 6 it forward me to the next question	I don't have to make any modify as it ben successful
2	To check the number of athlete participating is valid or not	Seven	The program wouldn't accept the result and will bring up error message	It shown an error as it wouldn't accept and word	I will not fix this error due to everyone likely to use number and it easier to use.
3	To check the number of athlete participating is valid or not	10	The program will not accept this result and it will show a message saying maximum 8 and minimum 4	The program displayed an error message saying the number must be between 4 and 8	According to the condition there must only be 4 to 8 participants
4	To ensure if the program accepts the finishing time of each athletes.	10.2	The program will accept this data because it is valid and the program will bring up the next question	The program did accept the data due to it's a valid data.	I will not have change any coding.
5	This will identify if the loop will stop on X number of data	5	The program will accept the number.	The loop finished at 5	It's been successful so I won't make any modification
6	This to check if the program will accept the gender that will be given by the users	Male	The program recognised the gender and it will process to next question	This is a normal data so the program will accept this	Again I won't have to change or modify anything.
7	This to check if the program will accept the gender that will be given by the users	Mle	The program will not accept any sort-cut, it will only process valid data which are	The program will not recognise this this data as it not coded in the program	The program will only accept valid text such as male or female

			male and female		
8	This is to ensure that all the records for woman are stored in the program	Normal Data	The program will recognise the record and will process the program forward	All the woman's records are recorded	They are stored as a variable
9	This is to ensure that all the records for man are stored in the program	Normal data	The program will recognise the record and will process the program forward	The program has stored all the records for man	Also stored as a variable
10	This is see if the program will proceed the end data from smallest to biggest	Normal Data	The program will process the data starting with the biggest and end with the smallest.	The end was in order starting with the biggest and stop	The program worked successfully as it was expected.

The learner has identified errors, but comments demonstrate a lack of understanding of the testing process.
Evidence of some errors being resolved is required for higher mark bands.

Mark in **band 1 (2 marks)**.

Activity 5

Activity 5: Evaluation

For this assessment, we were tasked with creating a simple program under the role of a Junior Software Developer for a local elite athletics academy. This program must be written to a brief set of specifications whilst ensuring a key few points:

- Standard programming conventions
- Efficient and robust
- Suitable data structures
- Outputs are meaningful
- Easy to use
- Tested using normal, abnormal and extreme data

The primary function of the program is to allow the user (in this case, an athletics coach) to record runner's times for a 100m race.
It must also allow the choice of gender for an athlete, allows the coach to enter times in lane order, displays finishing times from fastest to slowest and also output a world record if an athlete has achieved one.
When beginning to design and develop this program, I worked with the intentions to make it as minimalistic and robust as possible whilst also achieving primary functions.
I began designing this project using standard programming planning methods such as a flowchart and pseudocode. Using a flowchart, I was able to clearly see how the inner functions of the program would operate and lay everything out before writing and developing the program entirely.
The flowchart has been created linearly using standard flowchart creation which clearly shows the user how the program would work through meaningful shapes. These shapes indicate a decision, process, input, output, etc.
Pseudocode was my next choice in designing the program before writing it. By creating a simple pseudocode document, it allowed me to see how I would go about writing and developing the program in Python (the chosen language for this assessment). My pseudocode included little coding conventions, making it easy to read and follow whilst also giving me a clear indication on how I would develop my program next.
When it came to writing the program using Python, I used my flowchart and pseudocode to begin. My coding style was to use functions and variables, both with meaningful names to allow myself and the reader/user a clearer understanding of how the program operates. I chose to use functions as I find them incredibly straight-forward and linear, removing any unnecessary confusion when writing my code as it breaks it up into neat sections.
Using functions, I split every task up individually, beginning with gender, lane and time. Within these functions, it allowed the user to input a gender, lane, and time and then add said time to a list.
After these three initial functions, I included three run functions. The first of the three run functions calls in the first three standard functions, allowing them to run independently when the run function is called. The second run

function is a run again function, giving the user the option to input more profiles or exit the program.
This is definitely needed as a minimum of four profiles should be entered at once as there is a minimum of four athletes in a race at once, with a maximum of eight. Inside this second function, the world record function is called too.
In the third run function, it calls the first run function four times before calling the second run function once. This forces the user to input four profiles before giving the option to quit. This is another reason why functions were used for this task, as for loops and range become obsolete when a function can be called simply, allowing any code within to be ran accordingly.
After these three run functions, the world record function has been declared. Within this, if a specific gender and time requirement is reached, the program will print out a world record depending on the athlete time. Unfortunately, I was unable to make my program display more than one world record which is something I will remember for my next project.
Standard programming conventions have been followed. Indentations have been put in place where necessary and meaningful and legible comments have been inserted above most pieces of code, showing the reader clearly what everything does. I have also chose to use the convention "camelCase" for my functions and variable names as this makes them clearer to read and easy to follow.
After the program was written, I created a test plan and then tested it a plethora of times using normal, abnormal and extreme data. I came to the conclusion that there are a few minute validation errors on user inputs but that is something I can include in my next project as I was unable to figure it out this time. Testing the program allowed me to see where I went wrong and what I can better but also it showed me how functional and efficient the program was.
In conclusion, I made few changes during the development process such as making the choice to use limited while loops instead of many as this prevents confusion and errors as I am more comfortable with functions.
Upon evaluation, I believe my project solution meets the requirements of the scenario as it is not only efficient but minimalistic, simple and robust.

The learner has demonstrated some understanding of technical concepts. Some valid justification has been made which lacks support of changes made during development. 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.

Mark in **band 2 (6 marks)**.

Evaluation

In this evaluation I will attempt to give all positive and negative views on the program throughout its creation and of the final product. In this evaluation I will cover: how well I met the requirements, the quality of the program, the choices made about the coding conventions and the changes made throughout the process

How well I met the requirements

In doing the task that I have been assigned I have not met all the requirements for the program. The program needed: to have a choice in whether their gender was male or female, be able to enter their finishing times display finishing times lowest to highest to two decimal places and a message should appear if one of the athletes has achieved a record. Out of four needs in the task assigned I have achieved one.

I have achieved in the need for the user to have a choice between male and female; my program allows users to input "m" or "M" for male or "f" or "F" for female. Once this is input into the text box and the "add the scores" button is pressed it rather assigns the variable gender to: male if "m" or "M" had been inputted or female if "f" or "F" had been inputted.

I have failed to achieve the need to be able to enter a finishing time for the user. This is due to an issue I encountered where the program would stop the code and display the message "you cannot store more than 8 athletes". Unfortunately I was unable to find the error in the code to be able to fix this issue.

Due to my inability to fix the issue with the user being able to input a score, this means that I was also unable to output the scores of the athletes onto the output rich text box. This is the third need I failed as it also means I was unable to place the score inputted to 2 decimal places and that I was unable to sort the array from highest to lowest.

The fourth need I have failed is to notify the user when someone has achieved a World, European or British championship record in time. This is due to an unexpected error that occurred when I was programming the male world championship notification. Whilst I attempted to fix this error I could find no solution and I made the decision to move onto other parts of the project instead of wasting more valuable time on writing more broken code.

In review of all the requirements it becomes clear to me that I have completely failing in accomplishing the task set forth for me. I will use this information to help prepare and get a better understanding of the C# language as to be more prepared next time.

Quality of the program

The program I have produced isn't of high quality; whilst I have used constant and correct naming conventions and I have laid out the program in a simple fashion the program still falls short. The program has a total of six errors. One of these errors stops the user from progressing any further in the program unfortunately even with my best efforts I was unable to fix this error even after deleting the code. This means that the program is unusable and even thou the gender is saved and some code may work the program stops from being used.

Choices made about coding conventions

When creating the program I decide to use a coding convention where I leave a "_" better each space in the program. This is due to the fact that in programming a variable cannot have a space and still declare it all as a variable

and it all needs to be connected in some fashion. I decided against using camel case which is laid out as "TallGjraffe" where's the selection I made would be laid out as "tall_giraffe" I used this consistently throughout all of my code.
I also commented on all code I had written this was done as to help anyone who's reading the code understand what I am attempting to do with the code at the very least. It also helps people follow along with what code I am writing and may help future programmers from looking at the code and editing it as it displays and segments to them each section of code.

Changes made throughout the process

Throughout making my code I have made many changes one of the more constant changes made would be changes in the name of variables. This would done for multiple reasons, one of the main reasons was to better suit the condition I was assigning them to. An example would be the score global variable whist the variable name itself didn't change its purpose did as originally it was to be assigned to the users text for the finishing times text box however I later changed this so that It was to be assigned to the total result of what was to be outputted.
I also changed many lines of code one being the gender code lines, originally they were set up as txt_gender.text = gender. However I later changed this format to if (txt_gender.Text == "M" || txt_gender.Text == "m") gender = "male"; this meant that the user couldn't just put any piece of text in and have that assigned to their gender and that It would only allow the gender male or female. However, this also meant that the user could only have male or female if they had inputted M, m, F or f this was due to me being unable to write code that could predict all possible outcomes of writing the word male or female. I also attempted to add a move event to the "add to scores" button to help users understand this, as they would move their cursor over the button and text would appear saying "Please make sure gender is typed with an F for female or M for male". This didn't work as I intended and this didn't serve in helping the user in any way without the final product working.

Summary

In summary I feel like I didn't well in the task set for me. Whist I achieved minor needs for the code I was unable to meet all the needs and I was unable to produce a final piece of code that worked with all the desired needs and the robustness that I intended. I will use this frailer in future instances to help ensure that I have the proper knowledge and resources for the future tasks set ahead for me.

The learner has demonstrated some accurate and relevant understanding of technical concepts. One valid justification has been made of changes made during development. Some valid and mostly supported justification of coding conventions used, and the learner has made some logical links between aspects of the solution and the requirements of the scenario even though most were not achieved.
Technical vocabulary used but does not support the arguments.

Mark in **band 2 (4 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. 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. The test plan must include normal, abnormal and extreme data. Testing must address errors encountered and how these were overcome.
- Ensure the Program uses accurate validation and 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.
- 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.

For more information on Edexcel qualifications, please visit
www.edexcel.com/quals

Pearson Education Limited. Registered company number 872828
with its registered office at Edinburgh Gate, Harlow, Essex CM20 2JE