eduresourceza.com

Downloaded from eduresourceza.com

Your School Details Here

nnn 2017

SUBJECT	:	INFORMATION TECHNOLOGY PRACTICAL
GRADE	:	12
TIME	:	3 hours
SET BY	:	Mr R Bradbury (IEB Cluster Educator)
MODERATORS	:	Cluster Educators
MARKS	:	120

IMPORTANT INFORMATION

- Your name must appear as a comment at the top of each Programming Unit/Class.
- Printing is only required once you have finished the examination. Make sure that programs are saved regularly in the designated directory.
- You are strongly advised to keep track of the time spent on tasks as it is easy to become bogged down on a task and then you may not have enough time to complete another task which may afford you more marks.
- It is obvious that a working program will receive good marks but don't become stressed if a program does not work properly. Marks are awarded for any and all parts of programs which are correct whether or not the program works entirely.
- You may not walk about or talk during the session.
- Save your submission documents and programs regularly! Extra time will only be allowed for the time lost in the event of a power failure.
- This paper consists of 8 numbered pages. When you are instructed to commence the examination, please check that the paper includes all the pages.

Downloaded from eduresourceza.com

Downloaded from eduresourceza.com

SECTION A DATABASES AND SQL

*You have been provided with a folder named **Section A**, containing the files **MTBSA.mdb** and **SQL Answer Sheet.rtf**. Please complete your answers in this folder.*

QUESTION 1 40 MARKS

The organisers of the annual calendar of mountain biking cross country races, **MTBSA**, keeps a database of the cyclists, their clubs, the annual races and results. The information has been stored in a database named **MTBSA.mdb**, in four tables: **tblCyclists**, **tblClubs**, **tblRaces** and **tblResults**. The design of each table and some rows of data are as follows:

tblCyclists

Field Name	Data Type
cyclistID	AutoNumber
cyclistName	Short Text
cyclistDOB	Date/Time
cyclistGender	Short Text
cyclistClub	Number

cyclistID	cyclistName	cyclistDOB	cyclistGend	cyclistClub
1	Budhai, Trishana	1986/02/26	f	6
2	Buys, Dylan	1983/04/15	m	4
3	Hendricks, Zoe	1995/08/16	f	1
4	Kgapola, Masego	1982/11/06	f	8
5	Minnie, Jaydene	1995/02/27	f	5
6	Moosajee, Aaminah	1994/01/12	f	6
7	Mpiti, Thabo	1995/04/11	m	9
8	Nekhavhambe, Itani	1988/10/13	m	5
9	Pinana, Maki	1986/03/13	m	7
10	Saunders, Jessie	1984/05/21	m	8

tblClubs

Field Name	Data Type
clubID	AutoNumber
clubName	Short Text
clubProvince	Short Text

clubID	clubName	clubProvince
1	Dark 'n Dirty	Gauteng
2	Durban Mountain Bike Club	Kwazulu Natal
3	Howick Mountain Bike Club	Kwazulu Natal
4	Hillbillies Mountain Bike Club	Western Cape
5	Johannesburg Mountain Bicycle Club	Gauteng
6	Rockhoppers Mountain Bike Club	Gauteng
7	Tygerberg MTB	Western Cape
8	Paarl MTB	Western Cape
9	Maverick Club Hillcrest	Kwazulu Natal

Downloaded from eduresourceza.com

Copyright © 2017

Downloaded from eduresourceza.com

tblRaces

Field Name	Data Type
raceID	AutoNumber
raceName	Short Text
raceDist	Number
raceDate	Date/Time

raceID	raceName	raceDist	raceDate
1	Durban Coastal Road Race	21	2016/01/10
2	In The Mountains - Cape Town	25	2016/04/16
3	Lowveld Annual	46	2016/09/24
4	Vaal Triangular	36	2016/11/26

tblResults

Field Name	Data Type
resultID	AutoNumber
cyclistID	Number
raceID	Number
cyclistTime	Short Text

resultID	cyclistID	raceID	cyclistTime
1	12	1	01:46:36
2	14	1	01:48:33
3	2	1	01:40:48
4	3	1	01:45:38
5	18	1	01:39:49
6	1	1	01:23:21
7	8	1	01:36:55
8	15	1	01:29:10
9	20	1	01:41:46
10	6	1	01:30:08

1.1 Write a query that will list all details of female cyclists in alphabetical order of their names. (3)

1.2 Write a query that will list the cyclists' names and dates of birth for all cyclists that were born in the months of March to October. (4)

1.3 Write a query that will list the names of all the cyclists that belong to the club 'Tygerberg MTB'. (4)

1.4 Write a query that will list the number of cyclists that belong to each province. The field headings must be **clubProvince** and **NumCyclists**. (5)

1.5 Write a query that will list the cyclists' names (in alphabetical order) who didn't participate in the last race (raceID = 4). (6)

1.6 The time for the cyclist in the race with resultID = 18 was entered incorrectly; it should have been 01:37:04. Write a query that will change the cyclist's time for resultID = 18. (2)

Downloaded from eduresourceza.com

Downloaded from eduresourceza.com

1.7 The organisers of the annual races would like to arrange 4 'Plate' races for cyclists who didn't qualify (at club level) to race in the 4 races already on the calendar. The races will take place at the same venues and on the same dates as the original races. Write a query that will add the new races to **tblRaces**, using the original data and naming the races as: 'Plate Race - ' + <original race name>. (7)

1.8 The final rankings of the 4 races are calculated by converting the string time for each cyclist into a number of seconds for each race and then adding the times for each race to get a total for each cyclist. The cyclist with the lowest total is the overall winner, and so on. Write a query that will list the winning three cyclists and their times. To be fair to the ranking system, only cyclists who completed all 4 races can be considered for the positions. (9)

Downloaded from eduresourceza.com

Copyright © 2017

Downloaded from eduresourceza.com

SECTION B OBJECT ORIENTED PROGRAMMING

*You have been provided with a folder named **Section B**, containing the file **Results.txt**. Please complete your answers in this folder.*

SCENARIO

Twenty five cyclists participate in the 'Grand Finalé Cross Country Mountain Biking Race' consisting of 4 legs (races).

A list of the 25 competitors entered into the race are stored in a text file called **Names.txt** with a competitor number which serves an ID, followed by their name, separated by a "#" character. The first 5 rows of the files are displayed below:

```
1#Budhai, Trishana
2#Buys, Dylan
3#Hendricks, Zoe
4#Kgapola, Masego
5#Minnie, Jaydene
```

Once the race was run, the text file called **Results.txt** stored the starting and finishing time of each of the four legs completed by a competitor. The competitor number, the leg number, start time and finish time were recorded. Note that:

- Competitors can complete the legs in any order
- All legs start on the hour between 08h00 and 13h00
- Some competitors did not participate in all 4 legs, possibly due to injury. In this case they will not have 4 entries into the Results.txt file.

The 1st 10 lines of the file **Results.txt** are shown below. The file is in the format: Cyclist ID, cyclist names, the leg completed, the start time and end time (in the format hh:mm:ss). The fields are separated by the "#" character.

```
8#2#12:00:00#17:53:16
17#2#13:00:00#18:17:31
18#4#10:00:00#12:41:16
3#4#08:00:00#12:55:30
3#3#08:00:00#11:31:26
3#1#08:00:00#11:08:30
25#1#11:00:00#17:30:30
12#4#11:00:00#15:14:20
10#3#09:00:00#13:12:00
14#3#13:00:00#19:31:50
```

INCLUDE UML – CLASS DIAGRAMS FOR QUESTION 2 AND 3

QUESTION 2 18 MARKS

Refer to the following UML diagram of the LegTime class.

LegTime
- Sting start - String end

Downloaded from eduresourceza.com

Downloaded from eduresourceza.com

+ LegTime(String start, String end)
+ strToIntTime(String s) : integer

2.1 Create a class called **LegTime** with 2 string properties to store the start and end times for a cyclist in a particular leg. (3)

2.2 Code a constructor to instantiate the 2 string properties, given 2 string parameters. (3)

2.3 Code a method named **strToIntTime** that will return a string time (hh:mm:ss), received as a parameter, into a total number of seconds.

For example, the string time '17:53:16' must have the 1st two characters '17' converted to an integer and multiplied by 3600 (hours to seconds). This must be added to integer value of the 4th and 5th characters '53' (multiplied by 60 (minutes to seconds)), which must be added to the integer value of the last 2 characters '16'.

This method must only be visible within this class. (5)

2.4 Code a method named **getLegTimeInSeconds** that will use the previous method to return the total time taken by the cyclist for this leg (end time in s minus start time in s).

This method must be visible outside of this class. (3)

2.5 Code a **toString** method for this class that will return a string with the following format:

'Start Time: ' + STime + <tab> + 'End Time: ' + ETime + <tab> + 'Total Time in Seconds:' + getLegTimeInSeconds + 's' (4)

QUESTION 3 **16 MARKS**

Refer to the following UML diagram of the **Competitor** class.

3.1 Code a class named **Competitor** to store the **ID** and **name** of a cyclist as well as an array of 4 **LegTime** objects. (3)

3.2 Code a constructor to instantiate the cyclist's **ID** and **name**. (2)

*Note: The elements of the array of **LegTime** objects cannot be instantiated at this stage as not all cyclists completed all legs. The **LegTime** objects will only be instantiated when the file data is read; hence, the uninstantiated **LegTime** objects will have a **NIL/null** value which can be used to generate appropriate outcomes later in the program.*

3.3 Add **accessor** methods to return the **name** and the **ID** of the cyclist (competitor). (1)

Downloaded from eduresourceza.com

Copyright © 2017

Downloaded from eduresourceza.com

3.4 Considering that the constructor doesn't instantiate any of the array objects, code a method, **setLegTimes**, that will instantiate a **LegTime** array element object by receiving 3 parameters: the 1st parameter must indicate which element of the array is to be instantiated (the leg, 1 to 4) and 2 further parameters to indicate the start and end times; the 2 times must be used to instantiate the **LegTime** array element (object). Note that the legs are numbered from 1 in the Results.txt text file. (3)

3.5 Code a **toString** method that will return the cyclist's data, using the following format:

```
Competitor ID: 6, Name: Moosajee, Aaminah...Results Summary:
Leg 1: Start Time: 09:00:00   End Time: 15:18:04      Total Time: 22684s
Leg 2: Start Time: 08:00:00   End Time: 11:29:50      Total Time: 12590s
Leg 3: Start Time: 10:00:00   End Time: 16:30:03      Total Time: 23403s
Leg 4: Start Time: 13:00:00   End Time: 15:04:58      Total Time: 7498s
```

The top line combines the ID and name properties of the competitor with the text "Results Summary:" The following lines include calls to the **toString** method of each of the **LegTime** array objects indicating the details of each Leg of the competitor's race.

The following example shows the output required if the competitor didn't get a result for a particular leg (i.e. the **LegTime** array object was not instantiated). The output must display '*** Not Completed ***' for that leg (Leg 2 in this example):

```
Competitor Id: 7, Name: Mpiti, Thabo...Results Summary:
Leg 1: Start Time: 12:00:00   End Time: 15:22:14      Total Time: 12134s
Leg 2: *** Not Completed ***
Leg 3: Start Time: 12:00:00   End Time: 17:25:09      Total Time: 19509s
Leg 4: Start Time: 08:00:00   End Time: 10:47:59      Total Time: 10079s
```

Add code to this **toString** method to check if the **LegTime** array object was instantiated, or not. (7)

QUESTION 4 **29 MARKS**

4.1 Code a class called **CompManager** to store exactly 25 competitors in an array of **Competitor** objects. (3)

4.2 Code a constructor to instantiate the array of **Competitor**s, using the text file **Names.txt**. The file name must be received as a parameter.

Note: When this class object is instantiated in the application (question 5), the **Competitor** array's elements will default to **NIL/null** values until they have been instantiated individually. (8)

4.3 Code a method named **toString** to display each competitor's results using the **toString** method of each **Competitor** array element, one below the other. Ensure that the heading "All Competitor Results:" is included. Note that the Legs are numbered from 1.

```
All Competitor Results:
Competitor Id: 1, Name: Budhai, Trishana...Results Summary:
Leg 1: Start Time: 10:00:00   End Time: 16:17:04      Total Time: 22624s
```

Downloaded from eduresourceza.com

Downloaded from eduresourceza.com

```
Leg 2: Start Time: 11:00:00   End Time: 14:59:03      Total Time: 14343s
Leg 3: Start Time: 10:00:00   End Time: 15:02:05      Total Time: 18125s
Leg 4: Start Time: 11:00:00   End Time: 13:33:29      Total Time: 9209s

Competitor Id: 2, Name: Buys, Dylan...Results Summary:
Leg 1: Start Time: 09:00:00   End Time: 12:27:31      Total Time: 12451s
Leg 2: Start Time: 10:00:00   End Time: 16:23:56      Total Time: 23036s
Leg 3: Start Time: 10:00:00   End Time: 12:53:11      Total Time: 10391s
Leg 4: Start Time: 11:00:00   End Time: 13:15:52      Total Time: 8152s

Competitor Id: 3, Name: Hendricks, Zoe...Results Summary:
Leg 1: Start Time: 08:00:00   End Time: 11:08:30      Total Time: 11310s
Leg 2: Start Time: 11:00:00   End Time: 15:27:01      Total Time: 16021s
Leg 3: Start Time: 08:00:00   End Time: 11:31:26      Total Time: 12686s
Leg 4: Start Time: 08:00:00   End Time: 12:55:30      Total Time: 17730s
...
```

The output for the 1st 3 elements is shown above. (4)

4.4 Code a method called **search** to find the position of a competitor in the array given the id of a competitor. Marks awarded for efficiency. (5)

4.5 Code a method called **addResults** to read the data in the text file Results.txt and instantiate the array of **LegTimes** for each competitor. Use the search method in 4.4 to help you find a competitor on the Competitor array given the competitor number read out of the Results text file. (9)

QUESTION 5 **3 MARKS**

5.1 Code an application class called **CompetitionUI** to instantiate a **CompManager** object. (1)

5.2 Add code to display all the cyclists with their details, using the **ListAllCompetitorResults** method of the **CompetitionManager** object. Since there are no results for any of the competitors, the competitors will be displayed with null as follows: (1)

```
Competitor ID: 1 Name: Budhai, Trishana  Results Summary
Leg 0: ***Not Completed***
Leg 1: ***Not Completed***
Leg 2: ***Not Completed***
Leg 3: ***Not Completed***

Competitor ID: 2 Name: Buys, Dylan  Results Summary
Leg 0: ***Not Completed***
Leg 1: ***Not Completed***
Leg 2: ***Not Completed***
Leg 3: ***Not Completed***

Competitor ID: 3 Name: Hendricks, Zoe  Results Summary
Leg 0: ***Not Completed***
Leg 1: ***Not Completed***
Leg 2: ***Not Completed***
Leg 3: ***Not Completed***
```

5.3 Call the **addResults** method to combine the results with each competitor and display the array. (1)

```
Results Added

Competitor ID: 1 Name: Budhai, Trishana  Results Summary
Leg 1: Start Time 10:00:00    End Time 16:17:04 Total Time: 22624s
```

Downloaded from eduresourceza.com

Copyright © 2017

Downloaded from eduresourceza.com

```
Leg 2: Start Time 11:00:00    End Time 14:59:03 Total Time: 14343s
Leg 3: Start Time 10:00:00    End Time 15:02:05 Total Time: 18125s
Leg 4: Start Time 11:00:00    End Time 13:33:29 Total Time: 9209s

Competitor ID: 2 Name: Buys, Dylan  Results Summary
Leg 1: Start Time 09:00:00    End Time 12:27:31 Total Time: 12451s
Leg 2: Start Time 10:00:00    End Time 16:23:56 Total Time: 23036s
Leg 3: Start Time 10:00:00    End Time 12:53:11 Total Time: 10391s
Leg 4: Start Time 11:00:00    End Time 13:15:52 Total Time: 8152s

Competitor ID: 3 Name: Hendricks, Zoe  Results Summary
Leg 1: Start Time 08:00:00    End Time 11:08:30 Total Time: 11310s
Leg 2: Start Time 11:00:00    End Time 15:27:01 Total Time: 16021s
Leg 3: Start Time 08:00:00    End Time 11:31:26 Total Time: 12686s
Leg 4: Start Time 08:00:00    End Time 12:55:30 Total Time: 17730s
```

Downloaded from eduresourceza.com

Downloaded from eduresourceza.com

QUESTION 6 **14 MARKS**

6.1 Add any methods and other code to any of the classes you have coded so far, to produce the output shown below. Marks for this question will be awarded for programming efficiency and use of object oriented programming principles. You must:

- Calculate the total time taken for all the legs for each competitor
- If a competitor didn't complete a leg, a penalty of 6 hours (21600s) must be added
- The overall times for each competitor must be displayed in seconds
- The overall winner with the lowest time must be displayed in the end.

Your output should appear as follows:

```
Results
Budhai, Trishana  64301s
Buys, Dylan  54030s
Hendricks, Zoe  57747s
Kgapola, Masego  58805s
Minnie, Jaydene  65531s
Moosajee, Aaminah  66175s
Mpiti, Thabo  63322s
Nekhavhambe, Itani  60145s
Pinana, Maki  79363s
Saunders, Jessie  77816s
Uwizeyimana, Faith  77296s
Xu, Zekun  68748s
Chhana, Thanvir  74772s
Ferreira, Lorenzo  69323s
Francke, Shenique  71307s
Johnson, Tyler  52417s
Khashane, Makhado  71541s
Maswanganyi, Kingsley  78402s
Moyo, Petronella  72747s
Payne, Josh  67693s
Phafuli, Pulumo  61070s
Sithole, Siphiwokuhle  69725s
Solomons, Iyanda  72550s
Vilana, Thandeka  78456s
Wolmarans, Jacques  81133s

The winner is Johnson, Tyler with a time of 52417
```

6.2 Add code to your application to display the results. (14)

Downloaded from eduresourceza.com

Copyright © 2017