



eduresourceza.com

SUBJECT	Information Technology	DATE	November 2022
GRADE	Grade 11	MARKS	150
EXAMINER	D Kench	DURATION	3 hrs
MODERATOR	IT Department		

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY:

1. This paper consists of **8** pages. Please check that your paper is complete.
2. Answer ALL questions.
3. Read each question carefully. Code according to the question.
4. Sample outputs have been provided for reference.
5. Put question numbers as comments in the program.
6. Save your work frequently. No additional time will be allocated if you are negligent in this regard.
7. If a piece of code is not working, comment it out and carry on with other questions. Uncomment it before submission.

Question	1	2	3	4	5	6	Total	%
Possible Marks	50	23	44	9	11	13	150	100
Actual Marks								

SECTION A

SQL

50 MARKS

All the pirates on a ship have decided to learn how to eat healthily. Their details are stored by their dietician in a table called **tblPirates** in the **Healthy** database. Their dietician has recorded, their full name, gender, weight, height, kilograms lost, their medical condition (if they have one), their role on the ship and their date of birth.

tblPirates table in **Design View**

Field Name	Data Type
PirateID	Number
Surname	Short Text
FirstName	Short Text
Gender	Short Text
Weight	Number
Height	Number
KgLost	Number
MedicalCondition	Short Text
Role	Short Text
DOB	Date/Time

Sample Data of **tblPirates**

PirateID	Surname	FirstName	Gender	Weight	Height	KgLost	MedicalCondition	Role	DOB
1	Bart	Big	M	105	1,93	12,4	High Blood ...	Mates	1976-10-14
2	Gertie	Green Eyed	F	80,2	1,3	31,2	Parrot Flu	Gunner	1977-04-20
3	Sparrow	Jack	M	74,3	1,64	12,4	Hangmans ...	Quartermas...	1978-02-17
4	Boris	One Handed	M	104,8	1,58	25,4	Amputee	Cook	1978-08-06
5	Jones	Davy	M	94	1,66	28,6	<NULL>	Gunner	1978-11-01
6	Craig	Capn	F	55	1,68	-5,9	Back Proble...	Captain	1982-02-18
7	Hook	James	M	85,1	1,62	13,6	Severe Scarri...	Sailmaker	1982-11-09
8	The Terrible	Yassi	F	58	1,84	0,6	Yardarm Jau...	Cook	1982-12-05
9	Arty	Armpit	F	109	1,81	13	Severe Scarri...	Mates	1985-07-25

QUESTION ONE

SQL

50 MARKS

Write SQL queries to do the following:

Paste the answers in the **SQL AnswerSheet.rtf** provided.

- 1.1 List all the male pirates sorted alphabetically according to surname. Display the **surname**, **gender** and **role** fields.

Correct output

surname	gender	role
Bart	M	Mates
Boris	M	Cook
Hook	M	Sailmaker
Jones	M	Gunner
Sparrow	M	Quartermaster

(4)

- 1.2 Display the female pirates who are either Gunners, Cooks or Mates.

Correct output:

PirateID	Surname	FirstName	Gender	Weight	Height	KgLost	MedicalCondition	Role	DOB
2	Gertie	Green Eyed	F	80,2	1,3	31,2	Parrot Flu	Gunner	1977-04-20
8	The Terrible	Yassi	F	58	1,84	0,6	Yardarm Jaundice	Cook	1982-12-05
9	Arty	Armpit	F	109	1,81	13	Severe Scarring	Mates	1985-07-25
12	Madmaster	McPirate	F	104	1,96	13,5	Hangmans Neck	Mates	1987-09-21
16	Will	Waterfowl	F	88	1,91	-9,6	Severe Scarring	Mates	1992-03-05
17	Swashbuckler	Silver	F	115	1,63	1,57	Hangmans Neck	Gunner	1993-02-10
20	Lost Boy	Lobster Legs	F	109	1,62	-6,5	Severe Scarring	Gunner	1999-01-11

(5)

- 1.3 Display the correct age in years of all pirates who do not have a medical condition. Display the first name, age and medical condition.

Correct output:

firstname	Age	MedicalCondition
Davy	43	<NULL>
The dreadful	36	<NULL>
One Leg Peg	33	<NULL>
The Ruthless	32	<NULL>
The Yellow Belly	23	<NULL>

(7)

- 1.4 Count the number of pirates who have not lost weight i.e. they have a negative value for the KgLost field. Name this field **NotLost**.

Correct output:

NotLost
4

(5)

- 1.5 Calculate the Body Mass Index (BMI) of all the pirates using the formula:

$$BMI = \frac{weight}{height^2}$$

Display the pirate's surname and BMI rounded to 2 decimal places. Sort the result with the highest BMI first.

Sample output:

surname	BMI
Gertie	47,46
Davey	45,34
Ratbreath	43,82
Swashbuckler	43,28
Boris	41,98
Lost Boy	41,53
Jones	34,11
Arty	33,27
Hook	32,43
Nautical	30,76

(6)

- 1.6 Create a code for each pirate using the last 3 letters of their surname, the second letter of their firstname and a random number between 10 and 99. Display the pirate's first name and their code.

Sample output: Note the last 2 digits will differ.

surname	firstname	Code
Bart	Big	arti78
Gertie	Green Eyed	tier57
Sparrow	Jack	rowa43
Boris	One Handed	risn33
Jones	Davy	nesa25
Craig	Capn	aiga16
Hook	James	ooka85
The Terrible	Yassi	blea98
Arty	Armpit	rtyr45
Davey	The dreadful	veyh11

(7)

- 1.7 List the pirates who are above average height. Display their surname and height.

surname	height
Bart	1,93
The Terrible	1,84
Arty	1,81
Madmaster	1,96
Oily Legs	1,84
Will	1,91
Nautical	1,72
Yellowbeard	1,75

(5)

- 1.8 There is a competition between the various roles on the ship based on how much each role has lost or gained as a group. Determine the total amount of weight lost or gained for each role.

Correct output:

Role	TotalWeightChange
Mates	29,3
Gunner	54,87
Quartermaster	19,7
Cook	26
Captain	-5,9
Sailmaker	13,4
Navigator	11,9
Cooper	26

(5)

- 1.9 The pirates are unhappy with the combined weight change for each role, as some pirates did not follow the eating plan. It is decided to award a special prize to a role based on the largest weight loss by a pirate in the role. Only consider roles whose greatest weight loss by a pirate is above 15.

Correct output:

Role	MaxWeightChange
Gunner	31,2
Cook	25,4
Cooper	18,8

(6)

SCENARIO

An events company manages many bands whose members are high school and university students. The events company keeps track of the performers (the members of the band) and their details.

Some performers have food allergies, and the events company must ensure they have EpiPens available to treat allergy emergencies at a festival. If a band has underage performers (under 18) the events company must provide one adult chaperone for each band.

You have been provided with the text file **performers.txt**.

Performers.txt stores the following details about each band member:

- band member's name in the format `firstName<space>surname`
- the Performer's band name
- a boolean value indicating whether they have a food allergy
- their date of birth.

A comma separates each piece of information. The information for each band member is on a new line

`name,bandName,foodAllergy,dob`

The first few lines are given below:

```

performers.txt - Notepad
File Edit Format View Help
Olivia Spencer,One Hundred Watermelons,FALSE,2004-03-16
Gugulethu Gcuma,Destiny's Grandchildren,FALSE,2003-06-27
Shirley Renwick,Odds and Seven Evens,TRUE,2005-10-09
Bindi Anthony,DMG Wannabes,TRUE,2005-04-03
Kuhle Mbongwe,Burnt Amber,FALSE,2008-05-23
Rafiq Varma,Desperately Thinking,TRUE,2005-08-16

```

QUESTION 2**23 MARKS**

Use the class diagram to code a **Performer** class.

Performer
- name : string - bandName : string - foodAllergy : boolean - dob : Date
+ Constructor(inName : string, inBandName : string, inFoodAllergy : boolean, inDob : Date) + getName() : string + getBandName() : string + getFoodAllergy() : boolean + getDob() : date + setBandName(inBandName : string) + toString() : string

- 2.1 Create a class called **Performer**. (1)
- 2.2 Create the **name**, **bandName**, **foodAllergy** and **dob** fields. Ensure the fields cannot be accessed from outside the class. (3)
- 2.3 Create a constructor to assign parameters to the fields of the class:
- **inBandName**, **inFoodAllergy** and **inDob** are assigned without modification
 - **inName** is in the format `firstName<space>surname`
 - Change the format to `Surname,<space>Initial`
 - For example, the name `Olivia Spencer` would be changed to `Spencer, O`
 - Assign the formatted name to the **name** field. (7)
- 2.4 Create accessor methods for the **name**, **bandName**, **foodAllergy** and **dob** fields. (4)
- 2.5 Create a mutator method for the **bandName** field. (2)
- 2.6 Create a **toString** method to return the values of the **Performer** object. The string must be returned in the following format:
- `name<space>"of"<space>bandName<space>"is allergic"<space>dob` – if the performer has a food allergy
- `name<space>"of"<space>bandName<space>"not allergic"<space>dob` – if the performer does not have a food allergy
- An example of both formats is below
- `Sibeko, T of Astralis Fanatics not allergic 2002-11-15`
- `Wei, Z of Astralis Fanatics is allergic 2006-08-14` (6)

QUESTION 3**44 MARKS**

- 3.1 Code a class called **PerformerManager**. (1)
- 3.2 Add an array called **pArr** that can hold 50 **Performer** objects and a **size** field that can track the size of the array. (4)
- 3.3 Add a constructor to read the data from **performers.txt**, create **Performer** objects and add them to the array called **pArr**.
- Scan and loop through the entire file.
 - Extract the data from each line.
 - Create a **Performer** object, add it to the array and increment the size counter.
 - Display an error message if the file cannot be found. (12)
- 3.4 Code a method called **searchPerformer**. This method should take in the Performer's name and return the player's integer position in the array called **pArr**. If the Performer is not found, the method must return -1. (6)
- 3.5 Some of the performers want to change to another band. Code a method called **changeBand** with two parameters: the Performer's name and the Performer's new band name. The method must return a boolean value of true if the Performer is found in the array and the name is changed. If the Performer is not found, false must be returned. Use the method you coded in Question 3.4 to help you.
- If the Performer was found the method should return true, otherwise, it should return false. (6)
- 3.6 Code a method called **sort** to sort the array according to the band's name. The order of each Performer within each band is irrelevant, provided the band names are sorted alphabetically from A to Z. (9)
- 3.7 Code a **toString** method. This method should return a string with the information of each **Performer** object in the array. Each member's information should be returned in their **toString** format. Each member must appear on a new line. The first line should contain the heading 'List of Performers'. (6)

QUESTION 4**9 MARKS**

- 4.1 Code an application class called **PerformerUI**, with a **main** method. (2)
- 4.2 Add code to instantiate a **PerformerManager** object. (1)
- 4.3 Add code to sort and display the performers. (2)
- 4.4 Add code to change the band of "Bhatt, G" to "Crossing the Animals". Display a message indicating whether the change was successful or not (4)

QUESTION 5**11 MARKS**

The event organiser needs to know how many performers need EpiPens and a list of their names.

In the **PerformerManager** class:

- 5.1 Add a method called **allergyList**. This method should take no parameters and return a string. One EpiPen is needed for each Performer who has a food allergy. Return a list of performers with food allergies under the heading "Performers with Allergies". Display the number of EpiPens needed at the end of the list.

Correct output:

```
Performers with Allergies
Wei, Z
Bhatt, G
Khanyeza, Y
Ntshangase, K
Anthony, B
Gambu, F
Sonani, V
Varma, R
Na, J
Mouse, M
Hlongwane, N
Renwick, S
Pencilbend, T
Glass, P
```

Provide 14 EpiPen(s). (9)

- 5.2 Add code to the **PerformerUI** to call the **allergyList** method in the **PerformerManager** class and display the list. (2)

QUESTION 6**13 MARKS**

A second list is needed of bands that need a chaperone. One chaperone is provided to a band if the band has one or more members under 18.

- 6.1 Add code for a method called **chaperoneNeeded** to the **PerformerManager** class to add the band's name to a list, if a band has one or more underage performers. The method must return a list of band names under the heading "Bands who need Chaperones". If a band has more than one underage performer, the band name must appear once on the list.

Correct output:

```
Bands who need Chaperones
Astralis Fanatics
Burnt Amber
Crossing the Animals
DMG Wannabes
Desperately Thinking
Destiny's Grandchildren
Odds and Seven Evens
Source and GO
The Moon Lords
```

(11)

- 6.2 Add code to the **PerformerUI** to call the **chaperoneNeeded** method in the **PerformerManager** class and display the list. (2)