education

Department:
Education
**PROVINCE OF KWAZULU-NATAL**

STANMORE SECONDARY SCHOOL

TRIAL EXAMINATION – 2020

GRADE 12

# INFORMATION TECHNOLOGY

PAPER 1 – PRACTICAL

TIME: 3 HOURS
EXAMINER: MR S. NAIDOO

MARKS: 150

This paper consists of 13 pages.

# INSTRUCTIONS AND INFORMATION

1. This paper is divided into FOUR sections. Candidates must answer ALL FOUR sections.
2. The duration of this examination is three hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.
3. This question paper is set with programming terms that are not specific to any particular programming language (Delphi).
4. Make sure that you answer the questions according to the specifications that are given in each question. Marks will be awarded according to the set requirements.
5. Answer only what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation.
6. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.
7. Routines, such as search, sort and selection must be developed from first principles. You may NOT use the built-in features of a programming language for any of these routines.
8. All data structures must be defined by you, the programmer, unless the data structures are supplied.
9. You must save your work regularly on the disk/CD/DVD/flash disk you have been given, or on the disk space allocated to you for this examination session.
10. Make sure that your examination number appears as a comment in every program that you code, as well as on every event indicated.
11. If required, print the programming code of all the programs/classes that you completed. You will be given half an hour printing time after the examination session.
12. At the end of this examination session you must hand in a disk/CD/DVD/flash disk with all your work saved on it OR you must make sure that all your work has been saved on the disk space allocated to you for this examination session. Ensure that all files can be read.

IMPORTANT FILES PROVIDED:

| Folder | Files | Purpose |
|---|---|---|
| Question1 | **Q1_P.dproj** | Main Project File (Open this file to answer Question 1) |
| | **sales.txt** | Text File used in Question 1.4 |
| Question2 | **Q2_P.dproj** | Main Project File (Open this file to answer Question 2) |
| | **clsClient.pas** | Object Class for Question 2 |
| Question3 | **Q3_P.dproj** | Main Project File (Open this file to answer Question 3) |
| | **salaries.mdb** | Microsoft Access 2000 Database File |
| Question4 | **Q4_P.dproj** | Main Project File (Open this file to answer Question 4) |

## SCENARIO
### *CAFE LYN's*

**Café Lyn's is a popular coffee shop in town. In addition to providing a wide variety of coffee and curries, it is highly popular because it provides Internet access to its patrons.**

**As part of their ongoing development, and with the growth of its customer-base, the little coffee shop requires new software to better manage its operations.**

## QUESTION 1 – GENERAL PROGRAMMING SKILLS

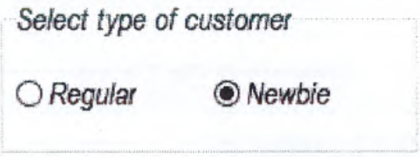
**Open data file Q1_P.DPROJ to answer this question.**

1.1 When a customer logs on to the order interface, a welcome message is displayed. The style of message differs depending on whether the customer is a "Regular" or a "Newbie".
***A Regular is a customer who visits the café often whilst a Newbie is a customer who is either new to the store; or only visits occasionally.***

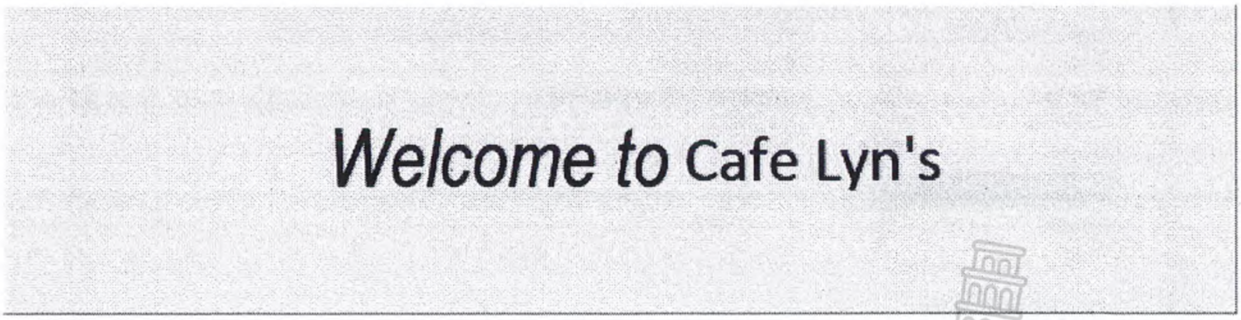
**Write code for the *rgpQ1_1Click* event which will:**

- Display "Welcome to Lyn's" in the Panel (*pnlMessage*)
- Extract and process the data from the Radio Group (*rgpQ1_1*) to determine if the user is a **Newbie** or **Regular**. Item (ItemIndex)
- **If the user is a Regular:**
  - Set the Panel (*pnlMessage*) Colour to White.
- **If the user is a Newbie:**
  - Set the Panel Font Style to Italic.

(5)


*Select type of customer*

○ *Regular* ● *Newbie*


*Welcome to* Cafe Lyn's

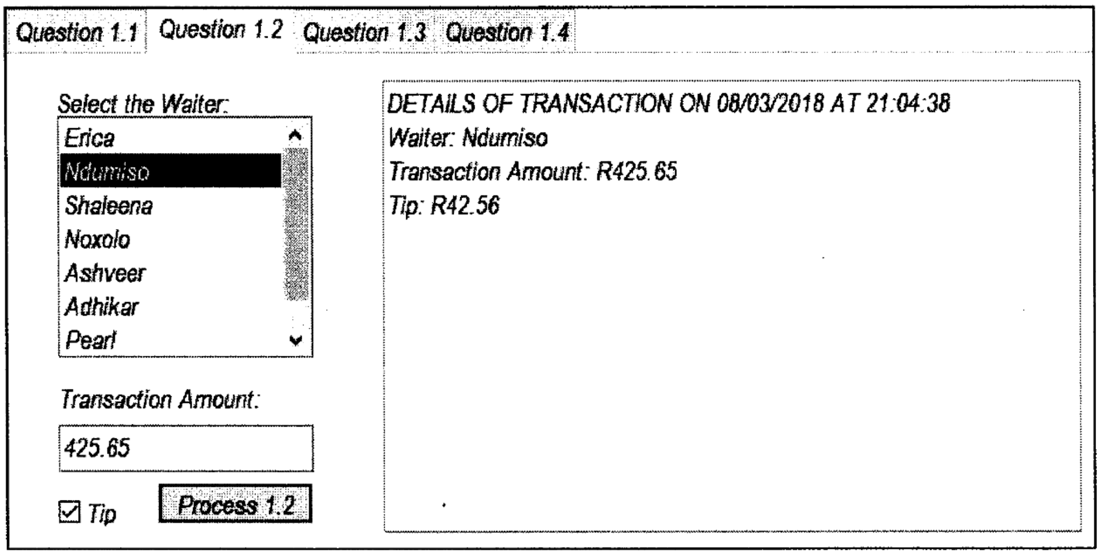
1.2 Waiters are responsible for tallying the final total of an order. A customer can choose whether to tip the waiter or not. Tips are standardized at 10% of the order value.

**Write code for *btn1_2* (Process 1.2) which will:**

- Extract the waiter's name from the List Box (*lstWaiters*).
- Extract the transaction amount from the Edit Box (*edtTrans*).
- If the Check Box (*chbTip*) is selected, calculate the tip value for the order.
- Clear the Output Area (*redQ1_2*).
- Display a Report in the Rich Edit Box (*redQ1_2*) in the following format:
  **DETAILS OF TRANSACTION ON [DATE] AT [TIME]**
  **Waiter: [Waiter Name]**
  **Transaction Amount: [Transaction Amount]**
  **Tip Amount: [Tip Amount]**

  *NOTE: The Date and Time must be extracted from the Operating System.*

**SAMPLE DATA #1**

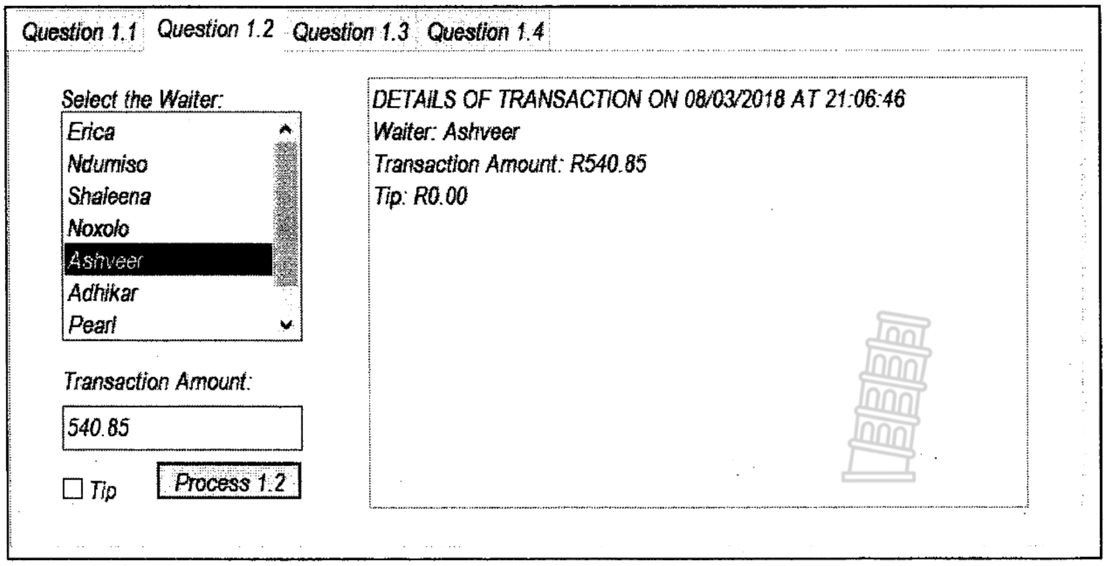


425.65

☑ Tip

```
DETAILS OF TRANSACTION ON 08/03/2018 AT 21:04:38
Waiter: Ndumiso
Transaction Amount: R425.65
Tip: R42.56
```

**SAMPLE DATA #2**



540.85

☐ Tip

```
DETAILS OF TRANSACTION ON 08/03/2018 AT 21:06:46
Waiter: Ashveer
Transaction Amount: R540.85
Tip: R0.00
```

(10)

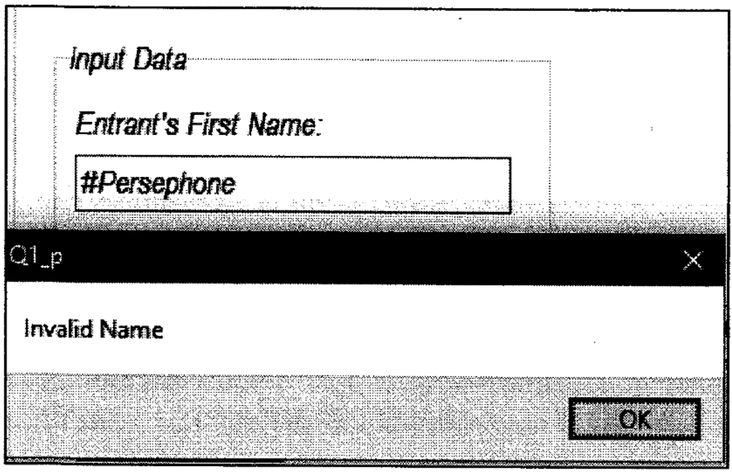
1.3 **As part of their marketing campaign, Café Lyn's is hosting a raffle for all customers. Each customer is allocated a generated entry code.**

**Write code for *btnValidate1_3* (Validate 1.3) which will:**

- Extract the entrant's name from Edit Box *edtEntrantName*.
- Validate the entrant's name. It should:
  - Contain letters of the alphabet only
  - Contain no spaces.

validate
val( _ , _ , check)

- If the name is invalid, display an appropriate error message using a Dialog Box.



Input Data

Entrant's First Name:

#Persephone

Q1_p

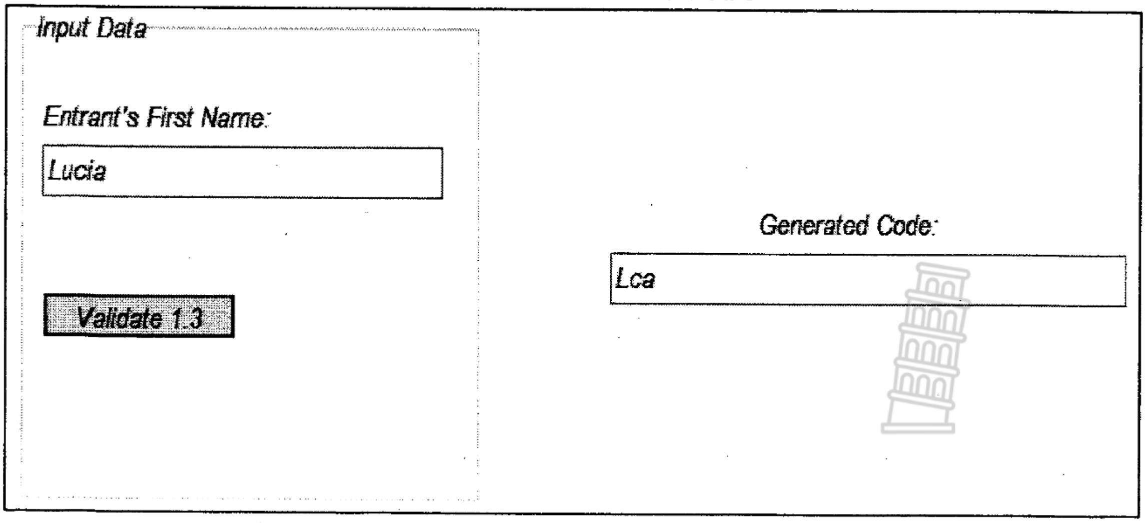
Invalid Name

val ('A..Z', sName, iCheck)
if check ≠ 0
AND
sLetter := copy(sName, iCount, 1)
if sLetter <> ' '
(inc

3 5 7 9

- If the name is valid, generate a user code for the entrant.
- The code should be generated as follows:
  - If the length of the name is an odd number:
    - The code should consist of the first, middle and last letter from the name.
  - If the length of the name is an even number:
    - The code should consist of the first 3 letters from the name reversed.
- The generated code should be displayed in Edit Box *edtGenCode1_3*.

**SAMPLE DATA #1**



Input Data

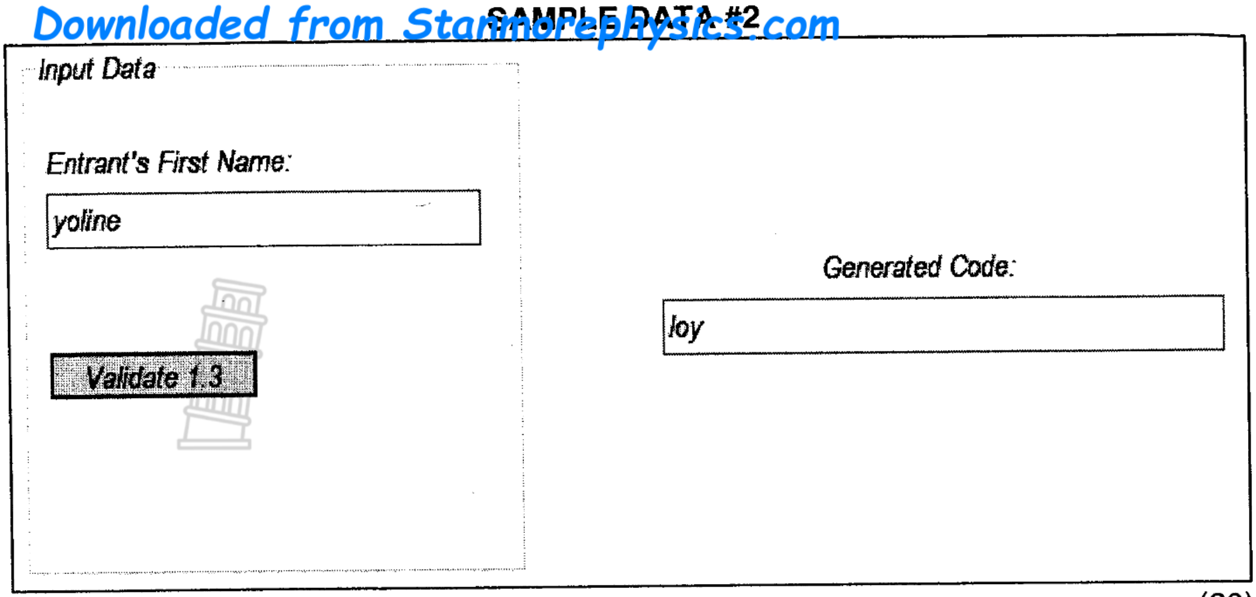
Entrant's First Name:

Lucia

Generated Code:

Lca

SAMPLE DATA #2



Input Data

Entrant's First Name:

yoline

Generated Code:

loy

(20)

**1.4 The auditors responsible for monitoring the income and expenses of Café Lyn's often require details on earnings for the current day.**

- Daily transactions are captured to a text file **SALES.TXT**.
- Each line in the text file has the total value of a single transaction.
- The system supports exactly 20 transactions a day. If there weren't 20 transactions on a particular day, the remaining transactions are captured with a total of zero (0).

The auditors may require information on a particular set of transactions – for example transactions 8 to 11. This "range" is specified using the Spinners **sedStart** (Start Transaction) and **sedEnd** (End Transaction).

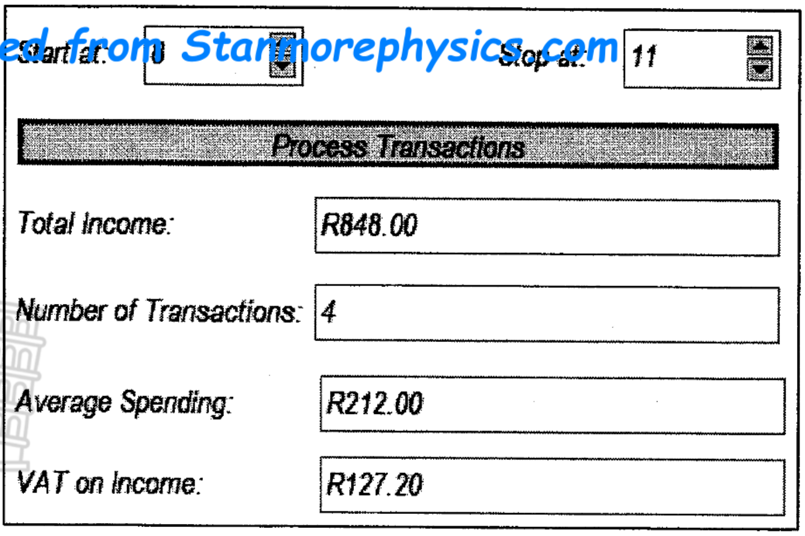
**Write code for *btnProcess1_4* (Process Transactions) which will:**

- Extract data from *sedStart* which indicates the transaction number (line number) to start calculations from.
- Extract data from *sedEnd* which indicates the transaction number (line number) to end calculations at.

- Connect to, and open text file **SALES.TXT**.
- Iterate/loop through the text file, reading and processing each line.

**Determine and display the following:**

- ***Total*** (sum of all transactions in the specified range) – Display in **edtTot1_4**.
- ***Number of Transactions*** – The number of transactions in the given range – Display in **edtNumTrans1_4**.
- ***Average Spending***: The average of all the transactions in the given range – Display in **edtAveSpend1_4**.
- ***VAT on Income***: The VAT payable on the total income displayed (15% of Total). Display in **edtVAT1_4**.



Total Income: R848.00

Number of Transactions: 4

Average Spending: R212.00

VAT on Income: R127.20

(15)

**SUBTOTAL: [50]**

## QUESTION 2 – OBJECT ORIENTATED PROGRAMMING SKILLS

**Open data files Q2_U.DPROJ and CLSCLIENT.PAS to answer this question.**

Being a coffee shop, Café Lyn's offers an Internet Hotspot to customers. Each client is granted a unique WiFi access code. Like all Hotspots, the free Internet is attached to a Fair Use Policy (FUP) which limits a user's transmitted data to a maximum of 1GB.
(1GB = 1024 MB)

To better manage the service, an Object Orientated application has been designed, but key functionality has not been coded.
You are required to write the necessary code in order to make the app function as described in the tasks that follow.

### 2.1 OBJECT CLASS (clsClient)

**2.1.1 Declare the following attributes for the object class using appropriate data types:**

| Name | Description | Example |
|---|---|---|
| fUserCode<br>String | A unique randomly generated 9 digit access code (grouped in 3's, separated by dashes) | 622-133-343 |
| fDataUsed Int | Data consumed by user in MB | 500 |
| fFUPActive<br>Boo | Whether, or whether not, the Fair-Use-Policy maximum use has been exceeded (FUP is exceeded when the user exceeds 1GB in use) | FALSE |

(4)

**2.1.2 Complete the code for the method GetUserCode which will:**

- Generate 3 random numbers in the range 100-999 (inclusive).
- Concatenate (join) the Random numbers, separating each number from the next using the dash symbol (-).

Random(999) +100
RandomRange(100,999)

**EXAMPLE:**
*First number*: 622
*Second number*: 133
*Second number*: 343

*Concatenated String*: 622-133-343

Assign the concatenated string to attribute **fUserCode**. (6)

**2.1.3** Complete the code for the mutator (setter) for **fDataUsed**. (1)

**2.1.4 Write a method called checkFUP which will:**

- Evaluate the current value of **fDataUsed**,
- If it has exceeded 1024, attribute **fFUPActive** should be set to **TRUE**.
- If it is at 1024 or below, attribute **fFUPActive** should be set to **FALSE**.

(4)

func

**2.1.5 Write code for the method *CalcDataUsed* which will return the user's currently used data as a String based on the following criteria:**

- Evaluate the attribute **fDataUsed**.
- If the used data is less than 500MB, it should be returned in MB with the unit (MB) at end.

MB ÷ 1024 = GB

**EXAMPLE:**
fDataUsed = 400
Return value: 400 MB

- If the used data ranges from 500MB to 1024MB, the value should be returned in GB. (Note: 1GB = 1024MB). The returned value should be rounded to 2 decimal places and include the unit (GB).

**EXAMPLE:**
fDataUsed = 600
Return value: 0.59 GB

- If the used data is over 1024MB, the word "CAPPED" should be returned.

**EXAMPLE:**
fDataUsed = 1025
Return value: CAPPED

(7)

**2.1.6 Complete code for the *toString* function which should return the attributes values formatted as shown below:**

User Code:	[fUserCode]
Usage:	[CalcDataUsed]

**EXAMPLE:**
User Code:	622-133-343
Usage:	0.59 GB

(3)

**2.2 This sub-question should be coded in the GUI Unit (Q2_U.PAS)**

**IMPORTANT NOTE**: As the User Code is randomly generated, your output will not necessarily match the screenshots.

Global object **objClient** has been declared.

**Write code for btnProcess:**

2.2.1 Instantiate global object *objClient* by calling its **Default Constructor**.

2.2.2 Call the method **GenUserCode** from the object class to generate a unique code for the user. ✓ ← UserCode

iDataUsed

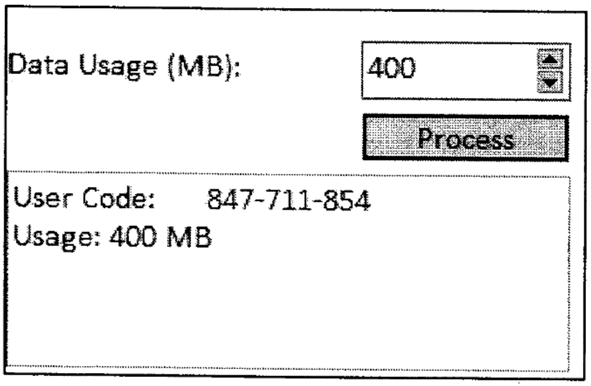
2.2.3 Obtain data from spinner (**sedDataUsed**) which indicates the current data used by client. Call the mutator for *fDataUsed* (*written in 2.1.3*), sending the data obtained from *sedUsedData* as an actual parameter.

2.2.4 Call method **checkFUP** from the object class.

2.2.5 Information on the object should be displayed to the Output Area (***redOut***) using the **toString** method. The output area must be cleared before displaying the information.

(5)

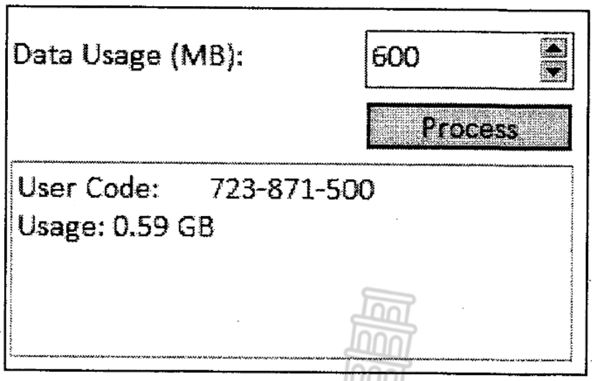
**SAMPLE DATA #1**



Data Usage (MB): 400

User Code:	847-711-854
Usage: 400 MB

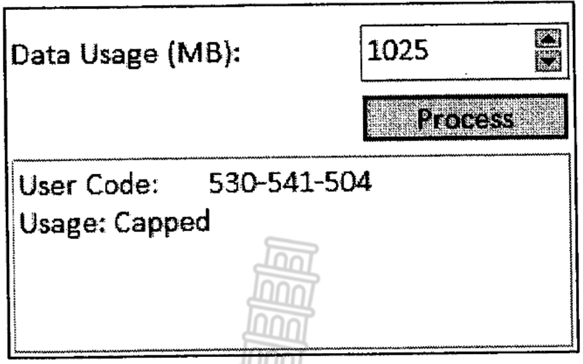
**SAMPLE DATA #2**



Data Usage (MB): 600

User Code:	723-871-500
Usage: 0.59 GB

SAMPLE DATA #3



Data Usage (MB): 1025

User Code: 530-541-504
Usage: Capped

**SUBTOTAL: [30]**

## QUESTION 3 – DATABASE MANIPULATION / SQLs

**Open Q3_P.DPROJ to answer this question.**

Each staff member's earnings is paid out once their commissioned income exceeds R500. The amount varies slightly depending on the individual transactions involved.

Details of staff earnings have been captured in a Database **SALARIES.MDB**.
The database has been connected to Delphi project **Q3_P.dproj** using the following ADO Components:

| | |
|---|---|
| **conStaff:** | ***ADOConnection component*** |
| **tblStaff:** | ***ADOTable component*** |
| **qryStaff:** | ***ADOQuery component*** |
| **dsStaff:** | ***DataSource component*** |

**NOTE**: The given code establishes a link to the database. Do not modify this code.

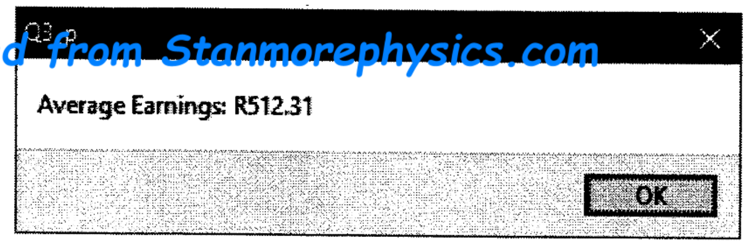
The database **salaries.mdb** has 1 table: **tblStaff** with the following fields:

| | |
|---|---|
| **StaffCode:** | ***Autonumber field which stores the waiter's staff number (Primary Key)*** |
| **Waiter:** | ***Text field which stores the waiter's name*** |
| **Gender:** | ***Text field of one character (M/F) which indicates Male/Female*** |
| **Earnings:** | ***Currency field which stores the waiter's earnings*** |

**3.1 The following questions must be answered using <u>CODE CONSTRUCT</u> only.**

**3.1.1 Café Lyn's needs to know the average earnings of its employees.**
**Write code for *btnCalcAve* (3.1.1 Calculate Average Earnings) which will:**

- Loop through **tblStaff**.
- Determine the total earnings and use this value to determine the average earnings.
- Display the average earnings (formatted as currency) using a Dialog Box.

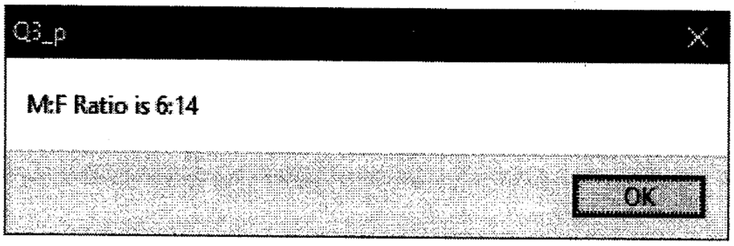


Average Earnings: R512.31

(12)

**3.1.2 The manager of Café Lyn's requires information on the gender demographic of the staff (the number of males vs the number of females).**
**Write code for *btnCountGen* (3.1.2 Determine Gender Demographics) which will:**

- Loop through **tblStaff**.
- Count the number of males and number of females
- Display the male:female ratio using a Dialog Box.



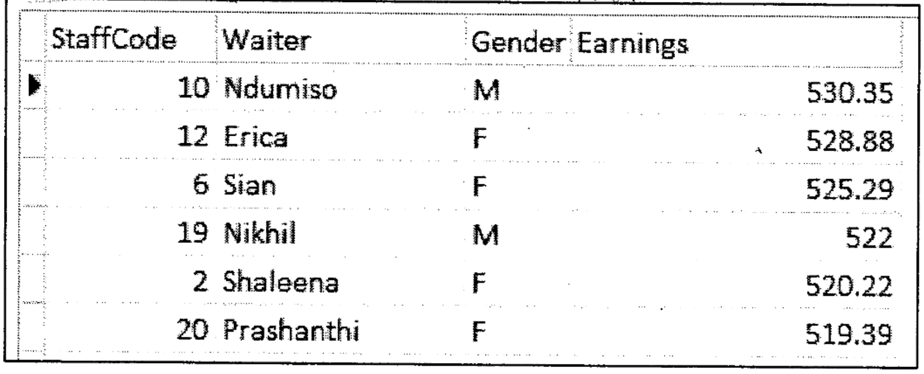
M:F Ratio is 6:14

$\frac{Num}{10}$ ×100

(12)

**3.2 The following questions must be answered using <u>SQL STATEMENTS</u> only. Assign variable sSQL in the relative button to your query.**

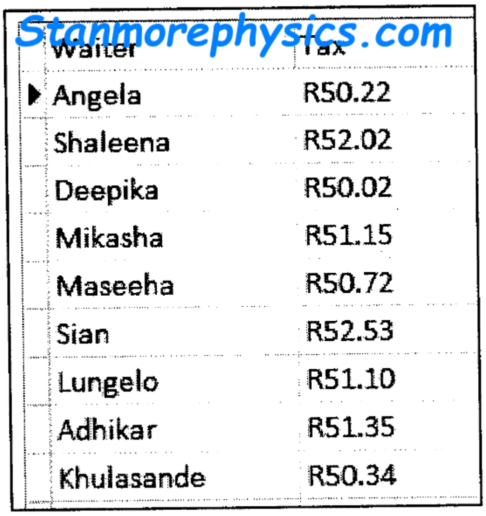
**3.2.1 In the code for *btnSortByEarnings* (3.2.1 Sort by Earnings):**
Write a SQL statement by completing the assignment statement for the variable sSQL to a query which will display all data from **tblStaff** sorted by the Earnings ORDER BY field, Highest to Lowest. DESC FROM (4)



| StaffCode | Waiter | Gender | Earnings |
|---|---|---|---|
| 10 | Ndumiso | M | 530.35 |
| 12 | Erica | F | 528.88 |
| 6 | Sian | F | 525.29 |
| 19 | Nikhil | M | 522 |
| 2 | Shaleena | F | 520.22 |
| 20 | Prashanthi | F | 519.39 |

**3.2.2 Employees are taxed at 10%. This deduction is effected by Café Lyn's before paying the employee.**
**In the code for *btnCalcTax* (3.2.2 Determine Tax on Income):**

- Write a SQL statement by completing the assignment statement for the variable **sSQL** to a query which will display the Employee's name and a calculated field called "Tax" which will contain the employee's tax amount (10% of earnings). The calculated field should be formatted as Currency. (7)



| Waiter | Tax |
|---|---|
| Angela | R50.22 |
| Shaleena | R52.02 |
| Deepika | R50.02 |
| Mikasha | R51.15 |
| Maseeha | R50.72 |
| Sian | R52.53 |
| Lungelo | R51.10 |
| Adhikar | R51.35 |
| Khulasande | R50.34 |

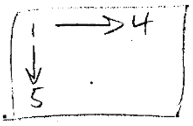
**NOTE**: In 3.2.1 and 3.2.2, you are ONLY required to write the SQL in the assignment of *sSQL*. Do not modify the existing code in buttons 3.2.1 and 3.2.2.

**SUBTOTAL: [35]**

## QUESTION 4 – ARRAYS

- Café Lyn's sells 5 different types of Coffee: **Brewed, Espresso, Cappuccino, Caffè Latte and Caffè Macchiato.**
- The sales (in cups sold) of each variant differs on a weekly basis, so it's difficult for the management to plan ahead when purchasing ingredients.
- Therefore a "Sales and Projections Portal" is needed. The GUI for this interface has been designed and some functionality has been added. However, key functionality is still missing.

### GIVEN STRUCTURES:



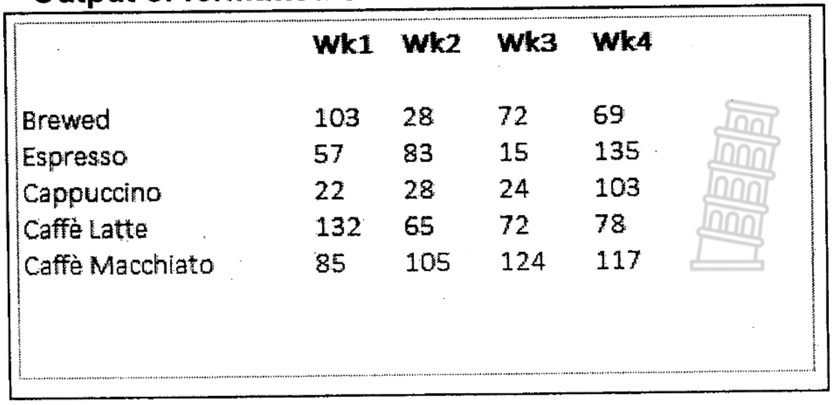
**arrSales**: A 2-Dimensional (2D) array with 5 rows and 4 columns. Each row represents a variant of coffee.
The columns represent sales for each of the 4 weeks of the current month.

**arrTypes**: A 1 Dimensional (1D) array with 5 elements storing the names of the variants of coffee available at Café Lyn's.

**popArray**: A method which displays *arrSales* it in Rich Edit Box *redMain*. It also provides the headings to Rich Edit Box *redOut*.

**NOTE**: Output in *redMain* and headings in both Output Areas have been hard-coded.

**Output of formatted Grid in Rich Edit Box *redMain*:**



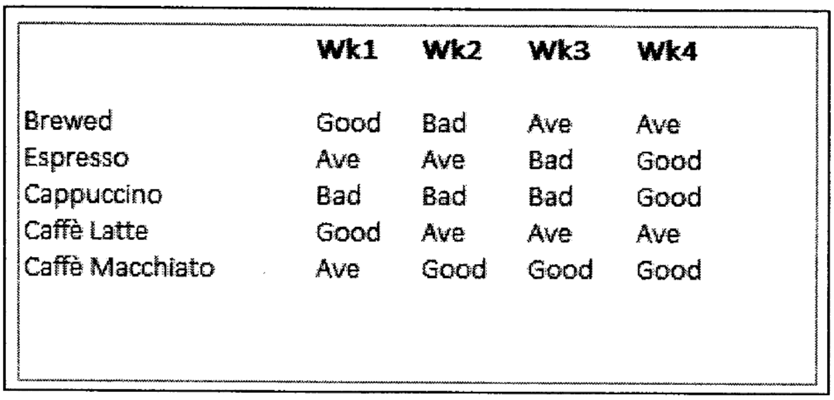
| | Wk1 | Wk2 | Wk3 | Wk4 |
|---|---|---|---|---|
| Brewed | 103 | 28 | 72 | 69 |
| Espresso | 57 | 83 | 15 | 135 |
| Cappuccino | 22 | 28 | 24 | 103 |
| Caffè Latte | 132 | 65 | 72 | 78 |
| Caffè Macchiato | 85 | 105 | 124 | 117 |

**4.1 The management at Café Lyn's does not like working with numbers and prefers that weekly sales be categorized into 3 levels:**

| **LEVEL RANGE** | |
|---|---|
| 0-49 cups sold: | **Bad** |
| 50-99 cups sold: | **Ave** (Average) |
| 100+ cups sold | **Good** |

**Write code for *btnSalesProcess4_1* (Process Sales) which will:**

- Loop through *arrSales*, evaluating the sales figures.
- Instead of displaying the sales value, output the corresponding level (Bad / Ave / Good) based on the Level Range above.
- Output should be displayed to Rich Edit Box *redOut*.
- Each row should start with the product name (extracted from *arrTypes*) as shown in the screenshot overleaf.



```
                     Wk1     Wk2     Wk3     Wk4

Brewed               Good    Bad     Ave     Ave
Espresso             Ave     Ave     Bad     Good
Cappuccino           Bad     Bad     Bad     Good
Caffè Latte          Good    Ave     Ave     Ave
Caffè Macchiato      Ave     Good    Good    Good
```

(12)

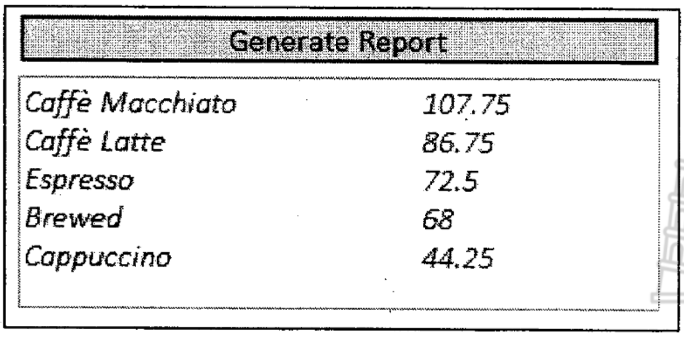
**4.2 The management of Café Lyn's needs to know the comparative popularity of each of its 5 variants of coffee. To achieve this, average weekly sales need to be calculated.**

**Write code for *btnGenReport4_2* (Generate Report) which will:**

Sum Row [illegible] [illegible] [c][1]

- Calculate and store the average weekly sales for each product. [illegible]
  (**NOTE**: *You have to define the necessary data structure to store the average sales for the 5 products.*)
- Sort the average sales in descending order.
- Display the coffee variant name and its corresponding sales average in descending order (Product with the highest sales at the top).

(23)



```
Caffè Macchiato          107.75
Caffè Latte              86.75
Espresso                 72.5
Brewed                   68
Cappuccino               44.25
```

**SUBTOTAL: [35]**

**GRAND TOTAL: 150**